

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Proyecto Fin de Carrera

Ingeniería Informática Superior

Diseño e Implementación de una Herramienta de Análisis de Registros basados en Snoopy para Linux

Tutor: Alejandro Calderón Mateos

Autor: María Yolanda González Castillo

*“El aprendizaje es experiencia...
Todo lo demás es Información.”*

Albert Einstein

Índice de Contenidos

Agradecimientos.....	17
Resumen	19
Abstract.....	20
1. Introducción.....	21
1.1 Motivación	21
1.2 Objetivos.....	24
1.3 Contribuciones al Estado del Arte	25
1.4 Plan de Trabajo	26
1.5 Estructura del Documento.....	28
2. Estado del Arte	31
2.1 Introducción.....	31
2.2 Sistemas Similares	35
2.2.1 Event Log Explorer	36
2.2.2 Performance Analysis of Logs (PAL)	41
2.2.3 Log Parser	45
2.2.4 Analog.....	56
2.3 Tabla Comparativa.....	61
Estudio de Viabilidad, Análisis, Diseño e Implementación.....	63
3. Estudio de Viabilidad del Sistema.....	63
3.1 Establecimiento del Alcance del Sistema	64
3.1.1 Estudio de la Solicitud.....	64
3.1.2 Identificación del Alcance del Sistema	66
3.1.3 Especificación del Alcance del EVS	67
3.2 Estudio de la Situación Actual.....	67
3.3 Definición de los Requisitos del Sistema	68
3.3.1 Identificación de los Requisitos	69
3.3.2 Catalogación de Requisitos.....	71
3.3.2.1 Requisitos de Usuario de Capacidad	71
3.3.2.2 Requisitos de Usuarios de Restricción.....	73
3.4 Estudio de Alternativas de Solución	75
3.4.1 Preselección de Alternativas de Solución	76
3.4.1.1 Alternativas de diseño	77

3.4.1.2	Alternativas de Lenguaje de Programación.....	78
3.4.1.3	Alternativas para la Base de Datos	79
3.4.2	Descripción de las Alternativas de Solución	80
3.5	Valoración de las Alternativas	82
3.5.1	Estudio de la Inversión.....	83
3.5.2	Riesgos y Planificación de las Alternativas	84
3.6	Selección de la Solución.....	85
4.	Análisis del Sistema de Información	87
4.1	Definición del Sistema	89
4.1.1	Marco Regulador	89
4.2	Establecimiento de Requisitos.....	90
4.2.1	Obtención de Requisitos.....	91
4.2.1.1	Requisitos Funcionales del Sistema	95
4.2.1.2	Requisitos No Funcionales del Sistema	97
	• Requisitos de Disponibilidad.....	97
	• Requisitos de Interfaz	99
	• Requisitos de Seguridad	101
	• Requisitos de Operación	101
	• Requisitos de Rendimiento.....	103
	• Requisitos de Restricción.....	104
	• Requisitos de Validación	107
4.2.2	Especificación de Casos de Uso	107
4.2.3	Análisis de Requisitos	116
4.2.3.1	Matrices de Trazabilidad	117
	• Requisitos de Usuario / Casos de Uso	117
	• Requisitos de Usuario / Requisitos Funcionales del Sistema	118
	• Requisitos de Usuario / Requisitos No Funcionales del Sistema	119
	• Requisitos Funcionales / Requisitos No Funcionales del Sistema	120
4.2.4	Validación de Requisitos.....	121
4.3	Identificación de Subsistemas de Análisis	121
4.3.1	Determinación de Subsistemas de Análisis	121
4.3.2	Integración de Subsistemas de Análisis	122
4.4	Análisis de los Casos de Uso	124
4.5	Elaboración del Modelo de Datos.....	125

4.5.1	Modelo Conceptual de Datos	125
4.5.2	Especificación de Necesidades de Migración de Datos y Carga Inicial.....	128
4.6	Definición de Interfaces de Usuario	128
4.6.1	Especificación de Principios Generales de la Interfaz.....	129
4.6.2	Especificación de Formatos Individuales de la Interfaz de Pantalla	130
4.6.3	Especificación del Comportamiento Dinámico de la Interfaz.....	131
4.6.4	Especificación de Formatos de Impresión	132
4.7	Aprobación del Análisis del Sistema de Información	133
5.	Diseño del Sistema de Información	135
5.1	Definición de la Arquitectura del Sistema	136
5.1.1	Definición de Niveles de Arquitectura	136
5.1.2	Especificación de Excepciones	139
5.1.3	Identificación de Subsistemas de Diseño	144
5.1.4	Especificación del Entorno Tecnológico	146
5.1.5	Especificación de Requisitos de Operación y Seguridad	147
5.2	Diseño de la Arquitectura de Soporte	148
5.2.1	Diseño de Subsistemas de Soporte	148
5.3	Diseño de Casos de Uso Reales	150
5.3.1	Identificación de Clases Asociadas a un Caso de Uso.....	150
5.3.2	Diseño de la Realización de los Casos de Uso.....	153
5.4	Diseño de Clases	161
5.4.1	Identificación de Clases Adicionales	161
5.4.2	Identificación de los Atributos y Métodos de las Clases	162
5.4.3	Diseño de la Jerarquía.....	186
5.4.4	Especificación de Necesidad de Migración y Carga Inicial de Datos	187
5.5	Diseño Físico de Datos	188
5.5.1	Diseño del Modelo Físico de Datos.....	188
5.5.2	Especificación de la Distribución de Datos	191
5.6	Generación de Especificaciones de Construcción	192
5.6.1	Especificación del Entorno de Construcción.....	192
5.6.2	Elaboración de las Especificaciones del Modelo Físico de Datos	193
5.7	Especificación Técnica del Plan de Pruebas.....	198
6.	Implementación.....	215
7.	Implantación	223

8.	Evaluación	225
9.	Planificación y Presupuesto	227
9.1	Planificación	227
9.2	Presupuesto	229
9.2.1	Descripción del Proyecto	229
9.2.2	Presupuesto Total del Proyecto.....	230
9.2.3	Desglose Presupuestario (Costes Directos)	230
9.2.3.1	Costes de Personal.....	230
9.2.3.2	Costes de Material	231
9.2.4	Resumen de Costes.....	232
10.	Conclusiones y Trabajos Futuros	233
10.1	Conclusiones	233
10.1.1	Producto	233
10.1.2	Proceso	235
10.1.3	Personales.....	236
10.2	Trabajos futuros.....	238
11.	Referencias Electrónicas y Bibliográficas	241
	Apéndices del Documento.....	247
12.	Manual de Implantación del Sistema	247
13.	Manual de Usuario	265

Índice de Ilustraciones

Ilustración 1: Cuota del uso de SSOO en Supercomputadoras.....	22
Ilustración 2: Cuota de Mercado de los SSOO de Escritorio.....	23
Ilustración 3: Ejemplo de Formato de Registro <i>Log</i>	32
Ilustración 4: Ejemplo de Registro <i>Log</i>	32
Ilustración 5: Ejemplo de un <i>log</i> estándar registrado con la herramienta Snoopy	32
Ilustración 6: Formato específico de los registros del fichero <i>log</i> que vamos a analizar	33
Ilustración 7: Ejemplo de un <i>log</i> específico del fichero que vamos a analizar	33
Ilustración 8: Event <i>Log Explorer</i> – Pantalla Principal	36
Ilustración 9: Event <i>Log Explorer</i> – Resultados Obtenidos	38
Ilustración 10: Performance Analysis of <i>Logs</i> (PAL) – Pantalla Principal.....	41
Ilustración 11: Performance Analysis of <i>Logs</i> (PAL) – Resultados Obtenidos.	42
Ilustración 12: <i>Log Parser</i> – Pantalla Principal.....	45
Ilustración 13: <i>Log Parser</i> – Resultados Obtenidos por pantalla.....	47
Ilustración 14: <i>Log Parser</i> – Resultados Obtenidos en un DataGrid.....	48
Ilustración 15: <i>Log Parser</i> – Resultados Obtenidos en un Gráfico de Barras 3D.....	48
Ilustración 16: <i>Log Parser</i> – Ejemplo de Consulta I	49
Ilustración 17: <i>Log Parser</i> – Resultado Mostrado I.....	49
Ilustración 18: <i>Log Parser</i> – Ejemplo de Consulta II	49
Ilustración 19: <i>Log Parser</i> – Resultado Mostrado II.....	50
Ilustración 20: <i>Log Parser</i> - Diagrama de formatos de entrada y salida.....	50
Ilustración 21: <i>Log Parser</i> – Resultados Obtenidos	54
Ilustración 22: Microsoft <i>Log Parser</i> GUI - <i>Log Parser Lizard</i>	55
Ilustración 23: Analog - Arranque y Lanzamiento de la aplicación.....	56
Ilustración 24: Analog - Índice para la navegación en un Informe	57
Ilustración 25: Analog – Resultados Obtenidos.....	58
Ilustración 26: Analog - Página de configuración de Informes.....	60
Ilustración 27: Actividades del Estudio de Viabilidad del Sistema de Métrica v.3	63
Ilustración 28: Actividades del Análisis del Sistema de Información de Métrica v.3	87
Ilustración 29: Diagrama General de Casos de Uso.....	108
Ilustración 30: Esquema de los Subsistemas de Análisis contemplados	123
Ilustración 31: Modelo Conceptual de Datos (Modelo Entidad/Relación).....	126
Ilustración 32: Actividades del Diseño del Sistema de Información de Métrica v.3	135
Ilustración 33: Niveles de Arquitectura	137

Ilustración 34: Niveles de Arquitectura con Subsistemas de Análisis	138
Ilustración 35: Arquitectura de Soporte del Sistema	149
Ilustración 36: Diagrama de secuencia: Caso de Uso 01 del Sistema de Información (CU-01)...	154
Ilustración 37: Diagrama de secuencia: Caso de Uso 02 del Sistema de Información (CU-02)...	155
Ilustración 38: Diagrama de secuencia: Caso de Uso 03 del Sistema de Información (CU-03)...	156
Ilustración 39: Diagrama de secuencia: Caso de Uso 04 del Sistema de Información (CU-04)...	157
Ilustración 40: Diagrama de secuencia: Caso de Uso 05 del Sistema de Información (CU-05)...	158
Ilustración 41: Diagrama de secuencia: Caso de Uso 06 del Sistema de Información (CU-06)...	159
Ilustración 42: Diagrama de Clases General del Sistema de Información	162
Ilustración 43: Diagrama Completo del Subsistema <i>Datos</i> - Componente Modelo.....	164
Ilustración 44: Diagrama Completo del Subsistema <i>Datos.Entidades</i> - Componente Modelo...	165
Ilustración 45: Diagrama Completo Subsistema <i>Datos.Acceso a Datos</i> - Componente Modelo	169
Ilustración 46: Diagrama de Clases Completo Subsistema <i>Presentación</i> - Componente Vista...	172
Ilustración 47: Esquema de Navegación entre Formularios.....	173
Ilustración 48: Diagrama de Clases del Subsistema <i>Operaciones</i> - Componente Controlador...	177
Ilustración 49: Diagrama Subsistema <i>Procesamiento Background</i> - Componente Controlador.	180
Ilustración 50: Modelo Relacional	191
Ilustración 51: Diagrama Distribución de Pruebas por Tipos y Módulos	199
Ilustración 52: Diagrama de Gantt con la Planificación de las Fases del Proyecto.....	229

Índice de Tablas

Tabla 1: Ejemplos de Ficheros <i>Log</i> según los datos de sus registros.....	31
Tabla 2: <i>Log Parser</i> – Ejemplos de Formatos de Entrada admitidos	46
Tabla 3: <i>Log Parser</i> – Ejemplos de Formatos de Salida	51
Tabla 4: Tabla Comparativa de las Soluciones Similares Propuestas	61
Tabla 5: Especificación del formato que seguirá nuestro catálogo de requisitos de usuario	70
Tabla 6: Requisito de usuario de capacidad 01 (RU-C01)	71
Tabla 7: Requisito de usuario de capacidad 02 (RU-C02)	71
Tabla 8: Requisito de usuario de capacidad 03 (RU-C03)	72
Tabla 9: Requisito de usuario de capacidad 04 (RU-C04)	72
Tabla 10: Requisito de usuario de capacidad 05 (RU-C05)	72
Tabla 11: Requisito de usuario de capacidad 06 (RU-C06)	73
Tabla 12: Requisito de usuario de restricción 01 (RU-R01)	73
Tabla 13: Requisito de usuario de restricción 02 (RU-R02)	74
Tabla 14: Requisito de usuario de restricción 03 (RU-R03)	74
Tabla 15: Requisito de usuario de restricción 04 (RU-R04)	74
Tabla 16: Requisito de usuario de restricción 05 (RU-R05)	75
Tabla 17: Requisito de usuario de restricción 06 (RU-R06)	75
Tabla 18: Comparativa Alternativas de Solución Propuestas.....	81
Tabla 19: Especificación del formato que seguirá nuestro catálogo de requisitos del sistema....	91
Tabla 20: Tipos de requisitos no funcionales contemplados	92
Tabla 21: Requisito Software Funcional 01 (RS-F01)	95
Tabla 22: Requisito Software Funcional 02 (RS-F02)	95
Tabla 23: Requisito Software Funcional 03 (RS-F03)	96
Tabla 24: Requisito Software Funcional 04 (RS-F04)	96
Tabla 25: Requisito Software Funcional 05 (RS-F05)	96
Tabla 26: Requisito Software Funcional 06 (RS-F06)	97
Tabla 27: Requisito Software No Funcional de Disponibilidad 01 (RS- NF-DI01)	97
Tabla 28: Requisito Software No Funcional de Disponibilidad 02 (RS- NF-DI02)	98
Tabla 29: Requisito Software No Funcional de Disponibilidad 03 (RS- NF-DI03)	98
Tabla 30: Requisito Software No Funcional de Interfaz 01 (RS-NF-IN01).....	99
Tabla 31: Requisito Software No Funcional de Interfaz 02 (RS-NF-IN02).....	99
Tabla 32: Requisito Software No Funcional de Interfaz 03 (RS-NF-IN03).....	100
Tabla 33: Requisito Software No Funcional de Interfaz 04 (RS-NF-IN04).....	100

Tabla 34: Requisito Software No Funcional de Interfaz 05 (RS-NF-IN05).....	100
Tabla 35: Requisito Software No funcional de Seguridad 01 (RS-NF-SG01).....	101
Tabla 36: Requisito Software No Funcional de Operación 01 (RS-NF-OP01)	101
Tabla 37: Requisito Software No Funcional de Operación 02 (RS-NF-OP02)	102
Tabla 38: Requisito Software No Funcional de Operación 03 (RS-NF-OP03)	102
Tabla 39: Requisito Software No Funcional de Operación 04 (RS-NF-OP04)	103
Tabla 40: Requisito Software No Funcional de Rendimiento 01 (RS-NF-RN01)	103
Tabla 41: Requisito Software No funcional de Rendimiento 02 (RS-NF-RN02).....	104
Tabla 42: Requisito Software No Funcional de Restricción 01 (RS-NF-RS01).....	104
Tabla 43: Requisito Software No Funcional de Restricción 02 (RS-NF-RS02).....	105
Tabla 44: Requisito Software No Funcional de Restricción 03 (RS-NF-RS03).....	105
Tabla 45: Requisito Software No Funcional de Restricción 04 (RS-NF-RS04).....	105
Tabla 46: Requisito Software No Funcional de Restricción 05 (RS-NF-RS05).....	106
Tabla 47: Requisito Software No Funcional de Restricción 06 (RS-NF-RS06).....	106
Tabla 48: Requisito Software No Funcional de Validación 01 (RS-NF-VA01)	107
Tabla 49: Formato para los Casos de Uso.....	109
Tabla 50: Caso de Uso 01 (CU-01)	110
Tabla 51: Caso de Uso 02 (CU-02)	111
Tabla 52: Caso de Uso 03 (CU-03)	112
Tabla 53: Caso de Uso 04 (CU-04)	113
Tabla 54: Caso de Uso 05 (CU-05)	114
Tabla 55: Caso de Uso 06 (CU-06)	115
Tabla 56: Trazabilidad entre Requisitos de Usuario Capacidad y Restricción y Casos de Uso	117
Tabla 57: Trazabilidad entre Requisitos de Usuario y Requisitos Funcionales del Sistema	118
Tabla 58: Trazabilidad entre Requisitos de Usuario y Requisitos No Funcionales del Sistema...	119
Tabla 59: Trazabilidad entre Requisitos Funcionales y No Funcionales del Sistema.....	120
Tabla 60: Tabla con los dominios de los atributos del modelo de datos	127
Tabla 61: Especificación del formato que seguirá nuestro catálogo de excepciones	139
Tabla 62: Excepción del Sistema 01 (EX_01)	140
Tabla 63: Excepción del Sistema 02 (EX_02)	140
Tabla 64: Excepción del Sistema 03 (EX_03)	141
Tabla 65: Excepción del Sistema 04 (EX_04)	141
Tabla 66: Excepción del Sistema 05 (EX_05)	142
Tabla 67: Excepción del Sistema 06 (EX_06)	142
Tabla 68: Excepción del Sistema 07 (EX_07)	143

Tabla 69: Excepción del Sistema 08 (EX_08)	143
Tabla 70: Clases Asociadas al Caso de Uso 01 del Sistema (CU-01)	150
Tabla 71: Clases Asociadas al Caso de Uso 02 del Sistema (CU-02)	151
Tabla 72: Clases Asociadas al Caso de Uso 03 del Sistema (CU-03)	151
Tabla 73: Clases Asociadas al Caso de Uso 04 del Sistema (CU-04)	152
Tabla 74: Clases Asociadas al Caso de Uso 05 del Sistema (CU-05)	152
Tabla 75: Clases Asociadas al Caso de Uso 06 del Sistema (CU-06)	153
Tabla 76: Especificación del formato de nuestro catálogo de componentes del Sistema	163
Tabla 77: Descripción del Componente de Diseño D-EN-01 (Clase Command)	165
Tabla 78: Descripción del Componente de Diseño D-EN-02 (Clase Category)	166
Tabla 79: Descripción del Componente de Diseño D-EN-03 (Clase QueryReport)	166
Tabla 80: Descripción del Componente de Diseño D-EN-04 (Clase UsedCommand)	166
Tabla 81: Descripción del Componente de Diseño D-EN-05 (Clase User)	167
Tabla 82: Descripción del Componente de Diseño D-EN-06 (Clase Constantes)	167
Tabla 83: Descripción del Componente de Diseño D-EN-07 (Clase Statistics)	167
Tabla 84: Descripción del Componente de Diseño D-EN-08 (Clase FilterDataResult)	168
Tabla 85: Descripción del Componente de Diseño D-EN-09 (Clase ActiveFilter)	168
Tabla 86: Descripción del Componente de Diseño D-AD-01 (Clase DatosBase)	170
Tabla 87: Descripción del Componente de Diseño D-AD-02 (Clase CommandDatos)	170
Tabla 88: Descripción del Componente de Diseño D-AD-03 (Clase CategoryDatos)	170
Tabla 89: Descripción del Componente de Diseño D-AD-04 (Clase QueryReportDatos)	171
Tabla 90: Descripción del Componente de Diseño D-AD-05 (Clase UsedCommandDatos)	171
Tabla 91: Descripción del Componente de Diseño D-AD-06 (Clase UserDatos)	171
Tabla 92: Descripción del Componente de Diseño D-AD-07 (Clase Sequences)	172
Tabla 93: Descripción del Componente de Diseño P-01 (Clase Form1)	174
Tabla 94: Descripción del Componente de Diseño P-02 (Clase Form2)	174
Tabla 95: Descripción del Componente de Diseño P-03 (Clase Form3)	175
Tabla 96: Descripción del Componente de Diseño P-04 (Clase IForm)	175
Tabla 97: Descripción del Componente de Diseño P-05 (Clase FormReferences)	176
Tabla 98: Descripción del Componente de Diseño P-06 (Clase FormUtilis)	176
Tabla 99: Descripción del Componente de Diseño O-01 (Clase InitialCharge)	178
Tabla 100: Descripción del Componente de Diseño O-02 (Clase FileTreatment)	178
Tabla 101: Descripción del Componente de Diseño O-03 (Clase Report)	178
Tabla 102: Descripción del Componente de Diseño O-04 (Clase ChargeData)	179
Tabla 103: Descripción del Componente de Diseño O-05 (Clase ProcessResult)	179

Tabla 104: Descripción del Componente de Diseño O-06 (Clase AppProcessException).....	180
Tabla 105: Descripción del Componente de Diseño O-W-01 (Clase IWorker)	181
Tabla 106: Descripción del Componente de Diseño O-W-02 (Clase AbstractWorker).....	181
Tabla 107: Descripción del Componente de Diseño O-W-03 (Clase DeleteDataWorker).....	182
Tabla 108: Descripción del Componente de Diseño O-W-04 (Clase GenerateReportWorker) ...	182
Tabla 109: Descripción del Componente de Diseño O-W-05 (Clase FileTreatmentWorker)	182
Tabla 110: Descripción del Componente de Diseño O-W-06 (Clase RestartDataBaseWorker) ..	183
Tabla 111: Descripción del Componente de Diseño O-W-07 (Clase SendEmailWorker)	183
Tabla 112: Descripción de Componente de Diseño O-W-08 (UpdateReportsFilterWorker)	183
Tabla 113: Descripción del Componente de Diseño O-W-09 (Clase RestartDataBaseWorker) ..	184
Tabla 114: Trazabilidad entre los Componentes y los Requisitos Funcionales del Sistema.....	184
Tabla 115: Trazabilidad entre los Componentes y los Requisitos No Funcionales del Sistema ..	185
Tabla 116: Definición de la Tabla Query Reports	187
Tabla 117: Tamaño de las tablas del sistema	190
Tabla 118: PL's y Secuencias específicos del sistema	196
Tabla 119: Formato General para el catálogo de Pruebas	199
Tabla 120: Prueba Unitaria 01 (PR_UN_01)	200
Tabla 121: Prueba Unitaria 02 (PR_UN_02)	200
Tabla 122: Prueba Unitaria 03 (PR_UN_03)	200
Tabla 123: Prueba Unitaria 04 (PR_UN_04)	201
Tabla 124: Prueba Unitaria 05 (PR_UN_05)	201
Tabla 125: Prueba Unitaria 06 (PR_UN_06)	201
Tabla 126: Prueba Unitaria 07 (PR_UN_07)	202
Tabla 127: Prueba Unitaria 08 (PR_UN_08)	202
Tabla 128: Prueba Unitaria 09 (PR_UN_09)	202
Tabla 129: Prueba Unitaria 10 (PR_UN_10)	203
Tabla 130: Prueba Unitaria 11 (PR_UN_11)	203
Tabla 131: Prueba Unitaria 12 (PR_UN_12)	203
Tabla 132: Prueba Unitaria 13 (PR_UN_13)	204
Tabla 133: Prueba Unitaria 14 (PR_UN_14)	204
Tabla 134: Prueba Unitaria 15 (PR_UN_15)	204
Tabla 135: Prueba Unitaria 16 (PR_UN_16)	205
Tabla 136: Prueba Unitaria 17 (PR_UN_17)	205
Tabla 137: Prueba Unitaria 18 (PR_UN_18)	205
Tabla 138: Prueba Unitaria 19 (PR_UN_19)	206

Tabla 139: Prueba de Integración 01 (PR_IN_01).....	206
Tabla 140: Prueba de Integración 02 (PR_IN_02).....	207
Tabla 141: Prueba de Integración 03 (PR_IN_03).....	207
Tabla 142: Prueba de Integración 04 (PR_IN_04).....	207
Tabla 143: Prueba de Integración 05 (PR_IN_05).....	208
Tabla 144: Prueba de Integración 06 (PR_IN_06).....	208
Tabla 145: Prueba de Integración 07 (PR_IN_07).....	208
Tabla 146: Prueba de Integración 08 (PR_IN_08).....	209
Tabla 147: Prueba de Integración 09 (PR_IN_09).....	209
Tabla 148: Prueba de Integración 10 (PR_IN_10).....	209
Tabla 149: Prueba de Sistema 01 (PR_SI_01).....	210
Tabla 150: Prueba de Sistema 02 (PR_SI_02).....	210
Tabla 151: Prueba de Sistema 03 (PR_SI_03).....	211
Tabla 152: Prueba de Sistema 04 (PR_SI_04).....	211
Tabla 153: Prueba de Sistema 05 (PR_SI_05).....	211
Tabla 154: Prueba de Aceptación 01 (PR_AC_01).....	212
Tabla 155: Prueba de Aceptación 02 (PR_AC_02).....	212
Tabla 156: Prueba de Aceptación 03 (PR_AC_03).....	213
Tabla 157: Prueba de Aceptación 04 (PR_AC_04).....	213
Tabla 158: Prueba de Aceptación 05 (PR_AC_05).....	213
Tabla 159: Prueba de Aceptación 06 (PR_AC_06).....	214
Tabla 160: Prueba de Aceptación 07 (PR_AC_07).....	214
Tabla 161: Prueba de Aceptación 08 (PR_AC_08).....	214
Tabla 162: Presupuesto – Descripción del Proyecto.....	229
Tabla 163: Presupuesto Total del Proyecto.....	230
Tabla 164: Costes de Personal del Proyecto.....	231
Tabla 165: Costes de Material del Proyecto.....	231
Tabla 166: Resumen de Costes del Proyecto.....	232

Agradecimientos

Este apartado me da la oportunidad de poder agradecer a todas aquellas personas que se han visto involucradas, de una manera u otra, en mi vida, en la consecución de mis éxitos, y en el aprendizaje de mis fracasos.

Todos estamos siempre rodeados de gente que nos enseña algo en la vida, y yo he tenido la oportunidad (y tengo, gracias a Dios) de seguir contando con el apoyo de muchos de ellos.

Me considero una persona bastante afortunada en lo que al destino se refiere, ya que en mi vida han pasado circunstancias tan dispares que una nunca hubiera pensado vivir donde vivo, estudiar donde he estudiado, trabajar donde trabajo, y conocer a la gente que conozco.

Sin lugar a dudas, la vida es un regalo que vale la pena vivir, sin pararte a pensar en cómo vas a acabar, dónde o con quién, porque ella sola se encarga de llevarte donde tu futuro te espera.

Por todo ello he de decir que estoy muy agradecida a todos los profesores que han pasado por mi vida, en especial a Alejandro Calderón, por hacerme ver que Linux no es un sistema operativo “poco convencional”, permitiéndome conocer todo el potencial que tienen y las mil y una maravillas que se pueden hacer con él.

A mis compañeros del camino académico, por las largas tardes estudiando, haciendo prácticas, y/o poniendo en común las dudas y problemas que no nos salían, porque verdaderamente me han enseñado que contar con un buen equipo es indispensable para llevar mejor los empedrados caminos de la vida.

A mi familia al completo, por confiar en mí y enseñarme que con fe y mucho trabajo podemos dar mucho más de lo que pensamos; por inspirarme a ser mejor cada día; por enseñarme que en la vida no todo es fácil y que hay que luchar por lo que se quiere; por ayudarme a superar los obstáculos que aparecen cuando menos te lo esperas; por ser mis compañeros más incondicionales;... en definitiva, ¡por todo! Porque sin ellos, la vida no sería lo mismo.

A mis padres, por estar siempre ahí, por ayudarme en todo momento, por hacerme quien soy, por guiarme en mi vida y ser mi luz.

A mis hermanos y sobrino... ¡por no dejarme estudiar tanto como creía necesitar! Dándole el punto de inflexión y diversión a mi vida.

A Manuel... por enseñarme que con paciencia y constancia, hasta ¡yo puedo ser la mejor mujer del mundo!

A todos aquellos que ya no están presentes, porque sé que si no fuera por ellos, no sería capaz de valorar todo lo que la vida me regala.

¡Gracias de todo corazón!

Resumen

El presente proyecto de fin de carrera, busca cubrir un vacío detectado en el conjunto de las herramientas de análisis de *logs* que conocemos actualmente.

La Universidad Carlos III de Madrid cuenta con varias herramientas instaladas en los equipos informáticos del campus que permiten a los administradores de sistemas monitorizar el uso que se hace de los ordenadores de las aulas por parte de los usuarios.

Una de estas herramientas es Snoopy [1], gracias a la cual se puede obtener un registro de todos los comandos ejecutados por un usuario desde la línea de comandos Linux de diversos equipos de la universidad.

Este registro, permitiría a los administradores de sistemas conocer un poco más acerca del uso que hace cada usuario de los ordenadores cuando trabajan con el sistema operativo Linux.

Pero, para poder llegar a tener este conocimiento, los administradores tienen que analizar los *logs* reportados por la herramienta de forma manual, con los inconvenientes que esto implica, no contando actualmente con ninguna herramienta que les ayude a realizar esta labor.

Nuestro objetivo en este proyecto ha sido justamente intentar cubrir este déficit, diseñando e implementando una herramienta que sea capaz de analizar los *logs* capturados con la herramienta Snoopy [1] y que ayude a los administradores de sistemas a realizar esta labor más fácilmente.

Abstract

The Thesis Project proposed, seeks to fill a gap found in the set of log analysis tools known today.

Working at the university, a system administrator has several tools installed on the computers of the campus that allow him to monitor the use made of classroom computers by users.

One of these tools is Snoopy [1], by which it can obtain a log of all commands used by a user, from the command line of Linux, during their session at any college equipment.

This record, allows system administrators to learn more about the use made by each user of the computer equipment of the university, especially when they work with the Linux operating system.

But to be able to have this knowledge, administrators have to analyze the logs reported by the tool manually, with the disadvantages that this implies not currently expecting any tool that helps them do this work.

Our goal in this project was just try to cover this deficit, designing and implementing a tool that analyze the logs captured with Snoopy [1] tool in order to help system administrators to perform this task easier.

1. Introducción

Podemos decir que el mundo en el que vivimos es, en sí mismo, una “fábrica inagotable” de información. Hecho que es particularmente evidente con la irrupción de internet y la aparición de numerosas compañías caracterizadas principalmente por la gran cantidad de datos que almacenan.

De hecho, bastaría prestar atención a lo que ocurre a nuestro alrededor para darnos cuenta ello. Cada vez es más común ver en nuestro día a día cómo el mundo tecnológico y los sistemas de información extraen datos de prácticamente todas las actividades cotidianas de casi cualquier persona, en cualquier lugar y en cualquier momento. Los cajeros automáticos de los bancos, las máquinas registradoras, las balanzas electrónicas de los supermercados, los teléfonos móviles, las *tablets* o los ordenadores, son claro ejemplo de ello.

Este proyecto se fija explícitamente en uno de esos casos: el uso de los equipos informáticos en la universidad.

El uso que hacen los alumnos de los ordenadores del campus genera, de manera automática e imperceptible por el usuario, una serie de registros log al alcance, mínimo, de los administradores de sistemas. Y, al igual que pasa con el uso de las cookies en los navegadores de internet, que permiten a las empresas responsables de los mismos utilizar la información obtenida de ellos para adaptar sus ofertas y/o productos a los usuarios, los administradores de sistemas de la universidad pueden utilizar la información obtenida del uso de los equipos para por ejemplo, mejorar el servicio prestado por los mismos y/o aumentar la seguridad de su uso.

Es aquí justamente donde cobran vital importancia los ficheros de registros *log* y las herramientas de para su análisis, una de las cuales es fruto del presente proyecto.

1.1 Motivación

¿Qué es lo que nos ha llevado, principalmente, a proponer y realizar este proyecto?

La principal motivación por la que nos hemos decidido a realizar el diseño y la implementación de una herramienta que ayude en el análisis de los ficheros log capturados con Snoopy [1] ha sido el gran potencial que podría llegar a tener toda la información contenida ellos.

Los administradores de sistemas de la universidad, por ejemplo, utilizan esta herramienta para monitorizar las órdenes que ejecutan, desde la línea de comandos de Linux, los usuarios de los equipos informáticos de las aulas; pero no cuentan con ninguna aplicación que les ayude a realizar este análisis de forma automática. Esto, hace que la tarea de análisis y abstracción de la información contenida en ellos se realice, principalmente, de forma manual; pudiendo llegar a convertirse en un proceso muy lento y pesado en función del tamaño del fichero y/o ficheros log que se requieran analizar.

Pero, este motivo de facilitación del trabajo no es el único que ha inspirado este proyecto, por cuanto la información contenida en los logs acerca del uso realizado por los usuarios de los distintos comandos Linux permite, además, detectar, en cierta manera, el nivel de destreza de los distintos usuarios en este sistema operativo.

Este dato, es una información clave para nosotros ya que entendemos que gracias a esta información los profesores y/o administradores de sistemas de la universidad, podrán proponer y orientar acciones destinadas a prestar al usuario de los equipos un mejor servicio de los mismos y/o dotar al sistema de una mayor seguridad. Lo que nos permite entender esta segunda motivación desde dos puntos de vista distintos, pero complementarios: poder medir la destreza de los usuarios con un sistema operativo tan potente [2] como poco expandido [3] como es Linux, y conseguir aumentar la seguridad de los equipos informáticos de las aulas de la universidad.

Por tanto, en lo que respecta al uso de los equipos, esta aplicación tiene como motivación principal facilitar a los administradores de los sistemas de la universidad, y a cualquier usuario interesado en ello, el análisis de los logs capturados con la herramienta Snoopy [1]. De ahí que el objetivo principal de este proyecto sea:

“Diseñar e Implementar una herramienta capaz de extraer automáticamente la información contenida en los registros *log* capturados con la herramienta Snoopy en un entorno Linux. “

Pero, con la consecución de este objetivo principal, se refuerza la posibilidad de conseguir también otros dos objetivos adicionales relacionados esencialmente con la forma en que un usuario utiliza los comandos Linux, y la seguridad del propio sistema.

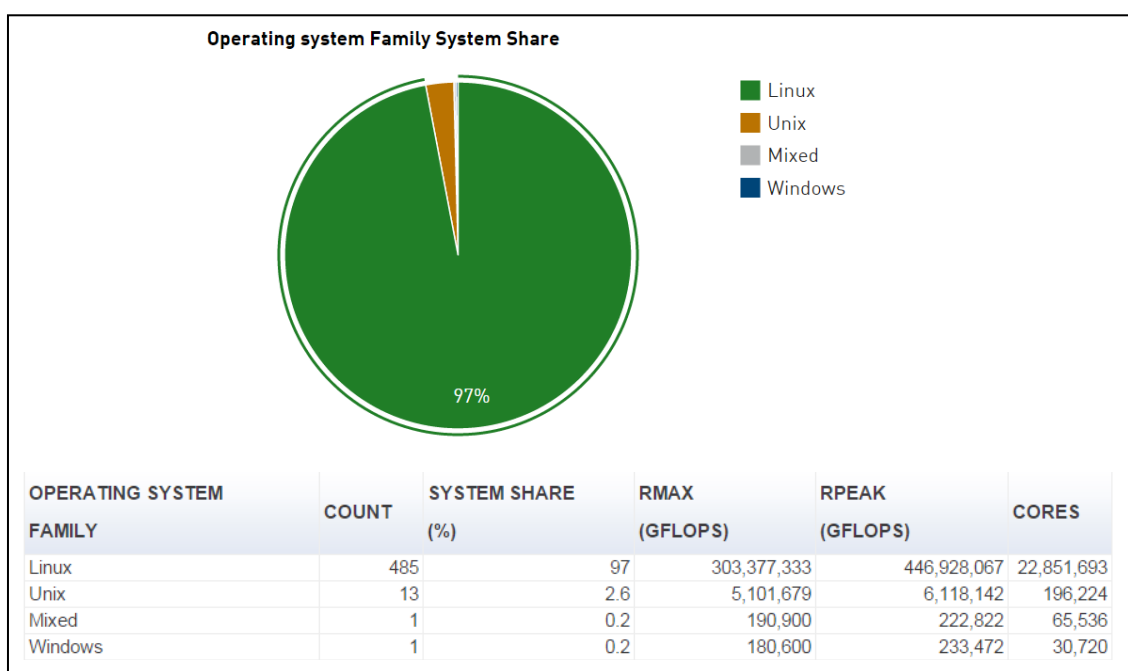


Ilustración 1: Cuota del uso de SSOO en Supercomputadoras.
Fuente: Top500 [2]

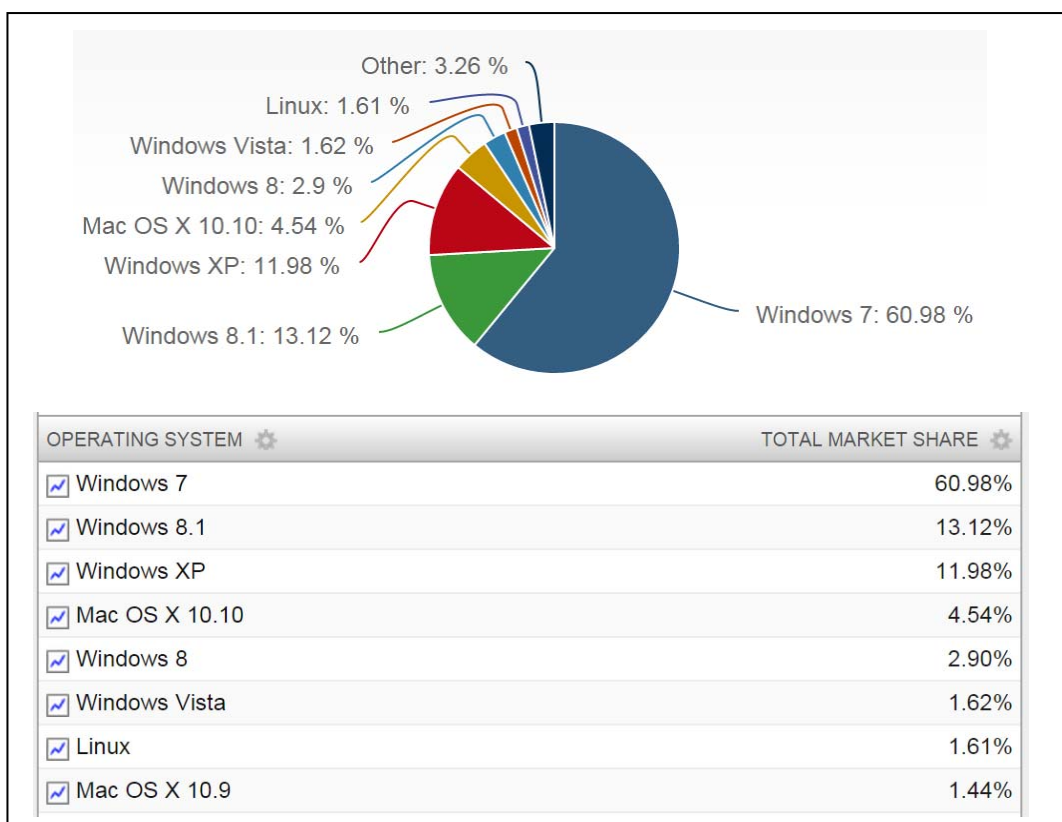


Ilustración 2: Cuota de Mercado de los SSOO de Escritorio.
Fuente: NetMarketShare [3]

Hoy por hoy, hacen uso de los equipos de la universidad una gran cantidad de personas, con perfiles y roles muy distintos, y lo habitual que sabemos de ellos es: su identificador de usuario, y el tiempo que han tenido su sesión abierta en el ordenador; no poniendo en valor, ni para el Usuario ni para la Universidad, los datos relativos a la forma en cómo usa los equipos.

Si un usuario fuese informado, o pudiese informarse, de cómo ha utilizado los equipos y cómo podría utilizarlos mejor, los resultados que obtendría serían más provechosos para él. Y, además, la forma como los usuarios usan los equipos informa también sobre la destreza que tienen para usarlos, lo que puede orientar, por parte de los profesores y/o administradores, acciones de mejora en el sistema para que el individuo pueda sacar el mayor partido posible de esas destrezas y desarrollarlas.

Esta información sobre la destreza es también básica para tomar medidas de seguridad, en cuanto a que el uso que se hace de los equipos informa no sólo de peligros en el sistema provocados por acciones de personas poco hábiles, sino también de aquellos peligros que puedan surgir con intencionalidad por parte de personas más hábiles.

Una herramienta que analizase en tipo real los comandos utilizados por los usuarios y determinase si se trata de un usuario potencialmente peligroso o no, permitiría a los administradores de sistemas prevenir posibles ataques y tomar las medidas oportunas en función del posible mal uso que se pudiera estar haciéndose del mismo.

Nuestra propuesta aquí expuesta, pretende justamente ser un prototipo de herramienta que, sin ser en tiempo real, realice este tipo de análisis con los logs capturados con la herramienta Snoopy [1], extrayendo la información necesaria para poder tipificar a los usuarios en función de la soltura con la que se desenvuelven en una línea de comandos tan poderosa y completa como la de Linux (que permite hacer prácticamente todo en éste sistema operativo), en orden a realizar mejoras que potencien y desarrollen su destreza en el uso de los equipos Linux y para una mayor seguridad del propio sistema.

De ahí que uno de los puntos principales de motivación para el presente proyecto sea la seguridad. Saber qué perfiles de usuario hay actualmente en las aulas, cuales podrían llegar a ser potencialmente peligrosos, o cuáles son totalmente inofensivos, es un punto clave del análisis de este tipo de ficheros de *log*.

Además, teniendo en cuenta que la universidad cuenta con un gran número de equipos conectados en la misma red y una gran variedad de usuarios con perfiles y roles muy distintos, esta idea de "monitorización" a nivel individual podría escalarse también a nivel de aula, edificio, o campus, inclusive.

1.2 Objetivos

De acuerdo a la anterior motivación, este proyecto busca conseguir, básicamente, tres objetivos:

Objetivo Principal

"Diseñar e Implementar una herramienta capaz de extraer automáticamente la información contenida en los registros log capturados con la herramienta Snoopy [1]."

Esto implica, que la aplicación resultante ha de permitir al usuario, y a los administradores de sistemas, realizar con ella las siguientes prestaciones:

1. Escoger el fichero de registros log capturados con la herramienta Snoopy que quiere analizar.
2. Proporcionar la lógica de análisis que quiere utilizar para analizar el fichero de logs facilitado.
3. Poder filtrar los datos de los que quiere obtener la información.
4. Seleccionar el tipo de análisis que quiere realizar, es decir, el tipo de informe que quiere obtener.
5. Obtener los informes de análisis realizados por la herramienta en formato Excel y PDF.
6. Enviar los informes generados, vía correo electrónico.

Objetivo Adicional

“Diseñar e Implementar una herramienta capaz de suministrar información sobre la soltura con que se desenvuelve un usuario en el uso de los comandos Linux, medida por el tipo de comando utilizado, su complejidad, dificultad o número de parámetros”.

Esto implica, que la aplicación resultante ha de permitir a los administradores del sistema realizar con ella las siguientes prestaciones:

1. Determinar los criterios de valoración de la destreza de los usuarios en el manejo de los comandos Linux y el peso de cada criterio contemplado, en orden a que puedan establecer los tipos de usuarios adecuados para sus finalidades.
2. Configurar nuevos tipos de informes a realizar actualizando la base de datos del sistema.

Puesto que uno de los objetivos a conseguir con el sistema es que los informes a obtener sean configurables según las necesidades del usuario, esta segunda prestación ha de realizarse de manera que la configuración de éstos sean lo más independiente posible de la aplicación, es decir, que la incorporación de nuevos tipos de informes no suponga prácticamente ninguna modificación en el conjunto del sistema, sino sólo una actualización en la base de datos.

Con todo, por razones de seguridad, esta prestación no se recomienda a todos los tipos de usuarios, sino solo a los administradores del sistema y a los usuarios o perfiles de usuarios que estos crean conveniente.

1.3 Contribuciones al Estado del Arte

En el capítulo destinado al Estado del Arte exponemos cómo las herramientas actuales diseñadas para el análisis de logs cubren un amplio abanico de posibilidades, pero no agotan todas las posibilidades de diseño existentes, en cuanto es imposible diseñar una única que sirva para todas las necesidades posibles.

Después de un repaso exhaustivo por prácticamente todo el estado del arte actual, si bien es cierto que se han encontrado multitud de herramientas distintas para el análisis de ficheros de *logs* de diferentes formatos, no hay ninguna específica en el mercado que sea capaz de cubrir todos los requisitos marcados en el sistema propuesto. Es decir, no hemos encontrado herramienta alguna que sea capaz de responder a todos los objetivos buscados y planteados en el apartado anterior, siendo la nuestra la única capaz de cumplir con los mismos.

La aplicación que presentamos, a diferencia de las existentes, está diseñada con la finalidad específica de

- Analizar un fichero de *logs* generado con la herramienta Snoopy [1].

Estos ficheros presentan sus registros en el siguiente formato específico:

<pre>Mes día HH:MM:SS nombre_maquina snoopy[id_snoopy]: [nombre_usuario, uid:número sid:número]: comando parámetros</pre>

Y, con estos datos, la aplicación realizará prestaciones similares a las que se pueden obtener con otras herramientas existentes en el mercado pero combinadas en una única y ofreciendo algunas nuevas:

- Tener una interfaz de usuario para facilitarle la interacción con la misma.
- Permitir la configuración de los informes por parte del usuario. Esto es, permitir al usuario que especifique los parámetros bajo los que quiere realizar el análisis, filtrar la información a obtener, seleccionar el tipo de informes a generar y proporcionar el fichero de *logs* que quiere analizar.
- Presentar los informes obtenidos de dicho análisis por pantalla.
- Proporcionar los resultados obtenidos en diferentes formatos para su posterior consulta o análisis
- Permitir el envío de los informes por correo electrónico.

Además, permite añadir (y modificar si se desea) información sobre los comandos (categoría, complejidad, dificultad y número de parámetros) en orden a poder apreciar a través de ella la destreza de los usuarios en el manejo de los comandos en Linux y su intencionalidad en el uso de los mismos.

Por ello, podemos decir que la contribución principal de este proyecto al estado del arte actual, no si no otra que el poder combinar todos estos factores anteriormente descritos en una única herramienta; ya que no hemos encontrado en el mercado ningún sistema implementado que aúne estas seis especificaciones buscadas, ni que sea capaz, simplemente, de analizar el tipo de fichero requerido con el formato y enfoque que se busca analizar.

1.4 Plan de Trabajo

Los pasos que hemos seguido para la ejecución del proyecto de fin de carrera aquí presentado, han sido los siguientes:

- 1) En primer lugar, tomando como punto de partida la motivación especificada en el apartado 1.1 y que tiene como base las inquietudes existentes con los *logs* en los equipos de la Universidad, nos marcamos los principales objetivos mencionamos en el apartado 1.2.

- 2) Una vez que tuvimos los objetivos suficientemente claros para iniciar la investigación, estudiamos el estado del arte del sistema propuesto, realizando un análisis exhaustivo de las herramientas que existían en el mercado en ese momento y que tenían un fin similar al que queríamos conseguir. Para ello utilizamos principalmente la bibliografía referenciada en el documento, utilizando ampliamente lo publicado en Internet sobre el tema.
- 3) En base a la información encontrada sobre las herramientas de análisis de logs bosquejamos una herramienta que, utilizando las ideas presentes en la herramientas estudiadas, diese una respuesta sencilla pero completa a la necesidad que nos planteamos al inicio del proyecto; e iniciamos el estudio de su viabilidad, estableciendo el alcance del proyecto, definiendo los requisitos del sistema buscado y analizando y valorando diferentes alternativas de solución posibles para su desarrollo e implementación.
- 4) Una vez determinado que el sistema era viable, redactamos de nuevo el catálogo de requisitos de usuario, comprobando que toda la funcionalidad requerida por el cliente (el tutor de este proyecto) estaba cubierta. Para ello, realizamos la especificación de los casos de uso, obteniendo a su vez nuevos casos de uso del sistema y aclarando el funcionamiento de algunas partes concretas del mismo.

Durante este Análisis del Sistema de Información, extendimos el catálogo de requisitos de usuario a un catálogo de requisitos software en los que se tuvieron en cuenta las necesidades específicas de software, hardware, recursos, interfaz, operación, seguridad, etc. (requisitos funcionales y no funcionales) según la alternativa de solución elegida para el desarrollo.

Identificamos los distintos subsistemas de análisis necesarios para la implementación del sistema, y llevamos a cabo la elaboración del modelo de datos específico para el correcto funcionamiento del mismo, estableciendo las entidades que queríamos tener presentes en él y las relaciones que debía existir entre ellas.

- 5) Acto seguido, realizamos el Diseño del Sistema de Información, definiendo los distintos niveles de la arquitectura del sistema, identificando los distintos subsistemas y diseñando las clases necesarias para la correcta implementación del sistema.

Con la arquitectura software bien definida, es decir, con la definición específica de cada subsistema, su funcionalidad principal, y las relaciones que iba a tener cada uno de ellos con los otros subsistemas de la aplicación, procedimos a elaborar un documento de planificación y presupuesto para el proyecto en desarrollo, y un plan de pruebas del sistema, es decir, un documento que nos ayudase a comprobar el correcto funcionamiento del mismo una vez lo tuviésemos todo implementado.

- 6) Una vez construida y probada la herramienta, procedimos a la redacción del manual de usuario y el manual de implantación del sistema, y el resultado lo sometimos a prueba en un grupo de “testers” que nos ayudaron a llevar a cabo el afinamiento de la herramienta.
- 7) Finalmente, una vez afinada la herramienta, la sometimos al plan de pruebas marcado para comprobar el correcto funcionamiento del sistema, realizando las correcciones pertinentes sobre los fallos encontrados y sometiendo el resultado al juicio del usuario con el que realizamos los ajustes finales.
- 8) Una vez obtenida la conformidad por parte del usuario, formalizamos el trabajo, redactando el documento actual al que añadimos nuestras conclusiones y abrimos camino a posibles nuevos trabajos futuros (avances) sobre el tema.

1.5 Estructura del Documento

En este apartado, se detalla la estructura del documento actual, indicando para cada capítulo los aspectos introducidos en él.

- En el capítulo 1, hacemos una breve introducción del proyecto, la **motivación** que nos ha llevado a proponer su realización, los **objetivos** que buscamos, y la **contribución** que supone su implementación al estado del arte actual. También explicamos brevemente el **plan de trabajo** seguido para su consecución.
- En el capítulo 2, presentamos el **estado del arte** actual. Analizamos lo que significa actualmente el análisis de log que nosotros queremos realizar y examinamos la forma como las actuales herramientas satisfacen necesidades similares a las nuestras; llegando a la conclusión de la conveniencia de elaborar una nueva herramienta que diseñada explícitamente para captar los datos contenidos en los registros log capturados con la herramienta Snoopy, reuniese en una sola herramienta las prestaciones de análisis e informe existentes en forma más o menos dispersa en otras herramientas, y añadiese información automáticamente relevante para tipificar a los usuarios y así poder realizar mejor las tareas de mejora y seguridad propios de la administración de sistemas. La base de tal conclusión se explicita en este capítulo donde se analizan explícitamente cuatro herramientas actuales representativas que nos han servido de base para dar un paso adelante y diseñar una herramienta más acorde con las necesidades de nuestro cliente.
- En los capítulos 3, 4 y 5, se lleva a cabo la realización de el proceso de *Desarrollo de Sistemas de Información* presentado en la estructura principal de Métrica v.3 [12], una metodología de desarrollo del software muy completa y conocida que ayuda a las organizaciones actuales a sistematizar las actividades que dan soporte al ciclo de vida del software.

- En el capítulo 3 se desarrolla el Estudio de Viabilidad del Sistema donde se especifica su alcance y los requisitos que se quieren cubrir con él. Además, se proponen distintas alternativas de solución, realizando una presentación de las mismas y mostrando las ventajas e inconvenientes que presenta cada una de ellas, de manera que al final de este capítulo, tengamos ya una propuesta de alternativa de solución elegida para el desarrollo.
- En el capítulo 4, y ya con la alternativa de solución elegida, se realiza el Análisis del Sistema de Información propuesto, donde se establecen los requisitos del sistema a desarrollar en base a los requerimientos marcados en el Estudio de Viabilidad del capítulo anterior y la alternativa de solución elegida. En este capítulo, se especifican además los casos de uso del sistema, los subsistemas de análisis de la aplicación y se analiza la trazabilidad de todos los requisitos buscados entre ellos y con los distintos casos de uso y subsistemas especificados, de manera que se compruebe la correcta consecución de las funcionalidades requeridas por el cliente. También se elabora el modelo de datos (entidad/relación) que deberá seguir nuestro sistema, especificando igualmente las necesidades de migración y carga inicial de datos de la aplicación. Además, se hace una pequeña definición de las interfaces de usuario que tendrá nuestro sistema, indicando sus principales funcionalidades y formatos.
- En el capítulo 5 se lleva a cabo la realización del Diseño del Sistema de Información en proyecto, cuyo objetivo principal es resolver el problema descrito y modelado en los capítulos anteriores. Para ello, se realiza la definición y diseño de la arquitectura del sistema propuesto y del entorno tecnológico en el que va a funcionar, detallando igualmente los algoritmos y técnicas concretas a utilizar. Además, se realiza una descripción detallada de los componentes del sistema, diseñando los casos de uso reales, las clases que vamos a implementar, y el modelo físico de los datos que vamos a necesitar. Como punto final de este capítulo, se realiza también la especificación técnica del plan de pruebas del sistema, indicando los distintos tipos de pruebas que se van a realizar sobre el mismo para comprobar su completitud y correcto funcionamiento.
- En el capítulo 6 se realiza una pequeña descripción de la implementación realizada, marcando únicamente aquellos aspectos que hemos considerado relevantes indicar en el presente documento, y añadiendo el código completo a un repositorio web desde donde se podrá descargar para una mayor información.
- En el capítulo 7 se marcan las pautas necesarias para la implantación del sistema en el equipo que se desee utilizar.
- En el capítulo 8 se detallan las técnicas de evaluación que se han realizado sobre el sistema implementado para comprobar su correcto funcionamiento, el cumplimiento de todas las funcionalidades requeridas, y el *feedback* obtenido de los usuarios (*testers*).

- En el capítulo 9 hacemos una presentación de la planificación del proyecto y del presupuesto que se requiere para la implantación y desarrollo del mismo.
- En el capítulo 10 se resumen las principales conclusiones obtenidas durante el desarrollo del presente proyecto, dividiéndolas bajo tres aspectos distintos: conclusiones acerca del producto obtenido, acerca del proceso que se ha llevado a cabo para su consecución, y conclusiones personales de todo el conjunto realizado. También se indica una serie de trabajos futuros que podrían complementar al sistema desarrollado.
- En el capítulo 11 se enumeran todas las referencias electrónicas y bibliográficas utilizadas durante la elaboración del presente documento.

Al final del documento, a parte de los capítulos descritos, se incluyen dos apéndices directos que sirven para aclarar y completar algunos aspectos específicos del proyecto desarrollado: el *Manual del Implantación del Sistema*, y el *Manual de Usuario*.

2. Estado del Arte

En este apartado se busca analizar qué sistemas hay actualmente en el mercado para “analizar un fichero de logs generado con la herramienta Snoopy [1]”, problema que se quiere resolver con el presente proyecto.

Pero, para poder abordar bien este tema, se ha considerado imprescindible hacer primero una pequeña introducción al análisis de logs [4], entiendo qué es un fichero de *logs*, en qué consiste el análisis de este tipo de ficheros, cómo se ha venido realizando hasta ahora y para qué se está utilizando, para poder encuadrar y entender más fácilmente el proyecto que tenemos entre manos.

2.1 Introducción

Qué es un fichero *log*

A efectos de este estudio, diremos que un fichero *log* es un fichero de texto plano cuyo contenido son los distintos registros *log* que va generando un sistema.

Hay infinidad de ficheros *log* en función del tipo de información que contengan sus registros. Veamos algunos ejemplos:

Fichero <i>Log</i>	Tipo de <i>Log</i>	Descripción
Error.log	Error	Almacenan mensajes de error producidos por el sistema. Es muy útil sobre todo para los desarrolladores, pues de otra manera les sería mucho más difícil “descubrir” qué ha pasado con el sistema en un determinado momento.
Access.log	Acceso	Registra los datos de acceso a un sistema
Sys.log	Sistema	Se encargan de registrar la información de lo que sucede en el sistema, sobre todo los mensajes de seguridad del mismo.
Info.log	Información	Sirve para registrar los distintos mensajes informativos que devuelve el sistema.
Debug.log	<i>Debug</i>	Almacena mensajes de depuración del programa.
User.log	Usuario	Registra información relativa al usuario
Event.log	Evento	Contiene todos los eventos acontecidos en un determinado sistema.
...*.log

Tabla 1: Ejemplos de Ficheros *Log* según los datos de sus registros

Estos ficheros, una vez actualizados con toda la información producida por el sistema, nos servirán para poder ver y analizar los distintos sucesos acontecidos en el mismo.

Todo registro que devuelve el sistema y que se escribe en el fichero *log* correspondiente, no sólo es relevante en cuanto al mensaje que contiene (mensaje de error, de seguridad, informativo, de usuario, etc.) sino que, para que sea realmente representativo y pueda proporcionarnos información completa del suceso acontecido, debe ir acompañado de otro conjunto de datos referentes al mismo que nos ayuden a entenderlo mejor. Por ejemplo, el mero hecho de recibir un mensaje que diga “error en el sistema A”, sin más información adicional, no es tan relevante ni informativo para el usuario como leer un mensaje completo de este estilo “se ha producido un error en el sistema A, el día 16 de octubre, durante la ejecución del proceso X, a las 16:30:05 hrs., estando activo el usuario Y”. Por ello, datos como la fecha (día, mes y año), la hora (indicando también minutos y segundos – y en algunos casos, hasta milisegundos), el proceso que lo produjo (nombre e identificador), el usuario que lo provocó (nombre y/o identificador), etc. suelen ser bastante útiles a la hora de completar la información producida del sistema y transmitirla al usuario final interesado.

El conjunto de datos que puede acompañar a un registro *log* para completar su información será igualmente distinto para cada tipo de *log* que queramos almacenar; volviendo a tener nuevamente infinidad de estructuras posibles para el almacenamiento de un mismo registro *log*, en función de los complementos que consideremos a bien añadir (aunque eso sí, es muy recomendable que dentro de un mismo fichero *log* se almacene únicamente registros del mismo tipo, y por tanto, con la misma estructura de datos, ya que de otra manera, se complicaría innecesariamente el tratamiento de los mismos).

Un ejemplo de formato de representación posible de un tipo de registro *log* podría ser:

`Mes Día Hora:Minutos:Segundos Hostname ProcessName [pid]: mensaje`

Ilustración 3: Ejemplo de Formato de Registro *Log*

Un ejemplo de un *log* registrado con este formato podría ser:

`Oct 01 16:30:05 A X [526501]: error en el sistema`

Ilustración 4: Ejemplo de Registro *Log*

En nuestro caso, el fichero de *logs* que vamos a analizar contiene registros *logs* capturados con la herramienta Snoopy [1].

Un ejemplo de registro típico capturado con esta herramienta es:

`2015-02-11T19:05:10+00:00 labrat-1 snoopy[896]: [uid:0 sid:11679
tty:/dev/pts/2 cwd:/root filename:/usr/bin/cat]: cat /etc/fstab.BAK`

Ilustración 5: Ejemplo de un *log* estándar registrado con la herramienta Snoopy

Pero, Snoopy [1] permite configurar el formato de sus registros en función de las necesidades del usuario, y de hecho, los *logs* del fichero que nosotros queremos analizar con la aplicación propuesta, siguen la siguiente estructura:

```
Mes día HH:MM:SS nombre_maquina snoopy[id_snoopy]:  
[nombre_usuario, uid:número sid:número]: comando parámetros
```

Ilustración 6: Formato específico de los registros del fichero *log* que vamos a analizar

Siendo un ejemplo de registro del fichero el siguiente:

```
Jan 20 06:25:08 horus snoopy[16611]: [unknown, uid:0 sid:16244]:  
/etc/cron.daily/sysstat
```

Ilustración 7: Ejemplo de un *log* específico del fichero que vamos a analizar

En qué consiste el análisis de un fichero *log*

Teniendo en cuenta que un fichero *log* es un archivo de texto plano donde se registran todos los sucesos acontecidos en un sistema; cabe entender que se necesite “algo” o “alguien” que se encargue de leer dichos datos almacenados en el mismo, y los transforme en información relevante para el usuario; es decir, se necesita “algo” o “alguien” que se encargue de realizar el análisis de dichos ficheros *log* y extraiga, de forma más o menos clara y automática, toda la información contenida en ellos. En esto consiste básicamente el análisis de un fichero *log*.

Muchas veces es tarea del propio administrador del sistema realizar esta labor, pero también le puede ser delegada a alguna persona que conozca bien el sistema, el fichero de *logs* que se quiere analizar y la estructura de sus registros, o incluso, especialistas externos que se dediquen a este tipo de análisis.

La información potencial que contienen estos ficheros *log* es tan grande y tan versátil que únicamente con un análisis y estudio exhaustivo y concreto de los datos almacenados en ellos, se puede llegar a obtener la información relevante contenida en los mismos.

Cómo se ha realizado el análisis de un fichero *log* hasta ahora

La complicación del análisis de un fichero *log* reside en que por su tamaño y el formato de los registros almacenados en ellos, estos ficheros suelen ser bastante difíciles y tediosos de manejar como para analizarlos a mano; lo que hace totalmente aconsejable el uso de una herramienta que se encargue de realizar un primer análisis de los datos almacenados en ellos y proporcionen al usuario un resumen de la información contenida en ellos en un formato amigable y mucho más fácil de interpretar por sí mismo.

En los ficheros *log* se almacena infinidad de datos que pueden ser utilizados para una gran diversidad de intereses, y no todos los datos son necesarios para todos los intereses, sino que se van adecuando a cada uno de ellos.

Es justamente por esta razón por la que los administradores de los sistemas no suelen leer los ficheros de *log* ellos mismos; sino que prefieren confiar esta labor a herramientas específicas de análisis ajustadas a las necesidades que tengan en cada momento. Herramientas que se encarguen de analizar todos los registros *log* almacenados en el fichero y que sólo les muestre la información que ellos consideran relevante analizar.

Dichas herramientas, en la mayoría de los casos, suelen ser herramientas hechas a medida por y/o para el administrador del sistema y el tipo de fichero que quiere analizar, o herramientas ya existentes acopladas a las necesidades del mismo.

Puesto que el tamaño de los ficheros *log* crece y crece constantemente con el tiempo, el programa o herramienta que utilicemos para analizar los registros almacenados en los mismos, ha de ejecutarse una o varias veces al día (en función del caudal de registro que se suceda en cada uno de los casos, evitando que se colapse o que llegue a un tamaño tal que sea impracticable trabajar con ellos), o incluso más veces si se quiere, ya que podría hacerse de forma automática sin tener necesidad de la supervisión del administrador.

De hecho, en la actualidad existen dos maneras principales de realizar el análisis de los ficheros de *log*:

- 1) Ejecutando el programa escogido por el administrador para realizar el análisis del fichero de *logs* de forma periódica, es decir, lanzándolo personal o automáticamente cada cierto tiempo de manera que se realicen comprobaciones controladas de los nuevos registros almacenados cada vez en el fichero;
- 2) Tener el programa escogido para realizar el análisis de nuestro ficheros *log* ejecutándose en el equipo constantemente, es decir, como un demonio que realicen las comprobaciones de nuestro fichero, registro a registro, *log* a *log*, según los vaya produciendo el sistema y se vayan almacenando en el mismo.

Esta decisión de elegir uno u otro método de ejecución del programa, nuevamente dependerá única y exclusivamente del administrador del sistema o de la persona encargada del análisis de los registros *log*; ya que es él el que mejor sabe cada cuánto tiempo es relevante analizar los datos introducidos en el fichero, cómo hay que hacerlo, y cuál es la capacidad de análisis que tiene la herramienta que ha decidido utilizar para tal fin. Aunque también, se podría tener en cuenta para esta decisión, cada cuánto tiempo hay que presentar el informe del análisis de los datos al cliente final interesado en la información contenida en los mismos, pues en función de esta necesidad, se podrán realizar análisis más o menos completos, en mayor o menor tiempo de ejecución.

Por todo ello, para poder realizar de forma coherente el análisis relevante en cada momento, antes de ejecutar el programa de escogido sobre nuestro fichero de *logs*, es recomendable que el usuario del sistema pueda definirle claramente y de manera precisa, sus propias reglas de filtrado del contenido y decirle qué es lo que desea exactamente analizar, dónde están los datos y cómo tiene que analizarlos, permitiéndole de esta manera tratar todos y cada uno de los registros contenidos en el fichero según el interés actual del usuario, y manteniendo en todo momento el resto de datos no tratados del fichero de *logs* intactos, de manera que puedan servir para un futuro análisis posterior con intenciones de recabar información distinta de los mismos.

2.2 Sistemas Similares

Una vez que ya hemos hecho una pequeña introducción de qué es un fichero *log*, en qué consiste el análisis de este tipo de ficheros, y cómo se ha venido haciendo hasta ahora, queda por ver, en el campo que compete al presente proyecto, ejemplos de sistemas similares de análisis de *logs* que puedan resolver el problema propuesto, de manera que podamos analizarlos y ver qué nos aporta cada uno.

Problema a resolver:

- Analizar un fichero de *logs* generado con la herramienta Snoopy [1] bajo una configuración específica.
- El formato de los registros almacenados en este fichero responde al siguiente formato:

```
Mes día HH:MM:SS nombre_maquina snoopy[id_snoopy]:  
[nombre_usuario, uid:número sid:número]: comando parámetros
```

- Queremos una herramienta que analice los *logs* almacenados en este fichero, y que muestre por pantalla al usuario los resultados del análisis.
- También, debe dar la opción de poder guardar los resultados obtenidos en diferentes formatos electrónicos para su posterior consulta o análisis, incluido el envío por correo electrónico.
- Además, el sistema ha de ser configurable por el usuario. Esto es, el usuario ha de poder especificar a la herramienta los parámetros bajo los que quiere realizar el análisis, indicarle el tipo de informes que quiere obtener, y poder seleccionar el fichero de *logs* que quiere analizar (todo siempre bajo el mismo formato).

En principio, **no hemos encontrado ningún sistema que sea capaz actualmente de resolver el problema propuesto bajo todos los requerimientos especificados.**

Pero, teniendo en cuenta que sí que hay varias herramientas que sirven para analizar ficheros de *logs* y mostrar y proporcionar los resultados obtenidos del análisis al usuario, se ha concretado el detalle de los requisitos buscados de la siguiente manera:

- La aplicación ha de ser capaz de Analizar un fichero *log* (vale cualquier formato);
- La aplicación debe mostrar los Informes obtenidos por pantalla;
- El usuario ha de poder elegir en qué formato (de entre al menos 2 posibles) quiere los informes resultantes del análisis;
- La aplicación ha de permitir el envío de los resultados del análisis por e-mail;
- El usuario ha de poder configurar los informes proporcionados: tipos de informe a obtener, parámetros para el análisis y datos que componen los informes resultantes.
- La aplicación debe contar con una interfaz de usuario que facilite la interacción con el sistema

Con estas nuevas especificaciones, se han vuelto a buscar sistemas similares que cumplan con los requisitos buscados, teniendo un mayor éxito que la vez anterior.

A continuación expondremos cuatro sistemas similares, seleccionados de entre todos los encontrados, por considerarse más populares, conocidos o extendidas actualmente.

Toda la información e imágenes utilizadas para la exposición de las distintas aplicaciones, han sido cogidas de la propia página web de la herramienta.

2.2.1 Event Log Explorer

Evento log Explorer [5] es una herramienta indicada principalmente para analizar los logs de eventos de un sistema operativo.

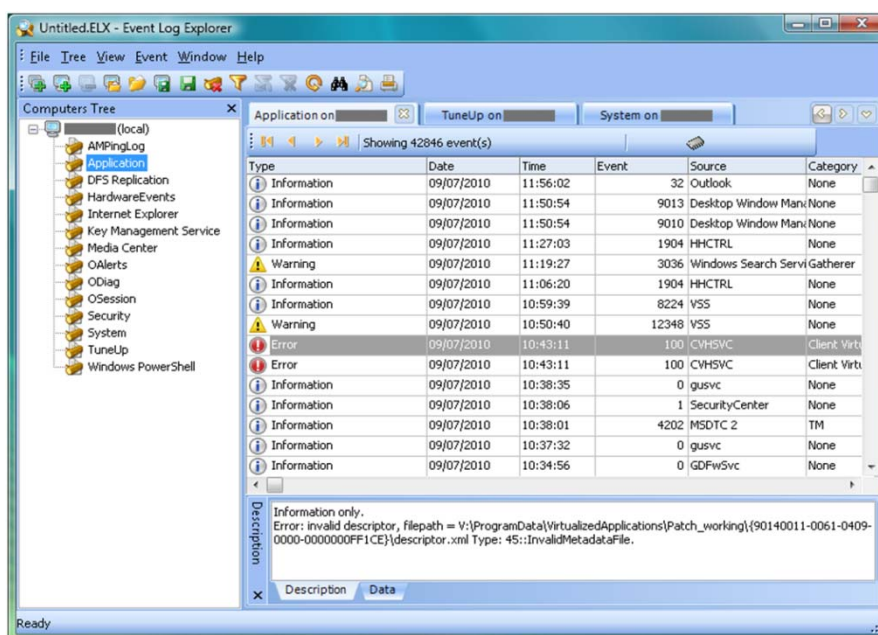


Ilustración 8: Event Log Explorer – Pantalla Principal
Fuente: eventlogxp [5]

RESUMEN DE LA APLICACIÓN

- Tiene licencia gratuita para uso personal, pero también existe la opción de comprar una licencia superior si se le va a dar una utilización de mayor alcance.
- Se creó fundamentalmente para analizar los *Logs* de Eventos de Windows, ampliando en gran medida las funciones de vigilancia estándar que presenta el propio Visor de Eventos de Windows y ofreciendo multitud de nuevas características que le permiten llevar a cabo el análisis de manera más eficaz: permite buscar, filtrar, imprimir y exportar los registros *log* del gestor de eventos de forma fácil y sencilla.
- Ya van por la versión 4.3 (lanzada el 8 de julio del 2014) y es compatible con prácticamente todas las versiones de Windows que se están utilizando actualmente (2000, XP, 2003, Vista, 2008, Windows7, Windows8).

CARACTERÍSTICAS BUSCADAS

a) La Aplicación ha de ser capaz de Analizar un Fichero *Log* ✓

¿Qué *logs* analiza? ¿Cuál es el formato de los datos de entrada?

Event *Log Explorer* analiza los ficheros *log* creados por el gestor de eventos del sistema operativo Windows.

Puesto que los archivos que analiza esta herramienta son los propios que produce el gestor de eventos del S.O. Windows, el formato de los registros *log* será el que venga reflejado en la API del mismo, admitiendo tanto el formato de registro de eventos clásico de Windows NT (archivos .EVT) como el nuevo formato de registro de eventos de Windows (archivos .EVTX).

La API moderna es algo más lenta, pero proporciona información más detallada acerca de los eventos.

b) La aplicación debe mostrar los informes obtenidos por pantalla ✓

Resultados mostrados por pantalla.

Para conseguir un análisis más efectivo de los eventos, y facilitar la interpretación de los resultados obtenidos, Event *Log Explorer* implementa también un conjunto de herramientas de informes que le permiten tener la opción de poder reportar los resultados del análisis en forma de informes analíticos con diferentes diagramas: gráficos de tarta y/o cuadros resumen dinámicos de los eventos; facilitando enormemente al usuario final interesado su interpretación y estudio.

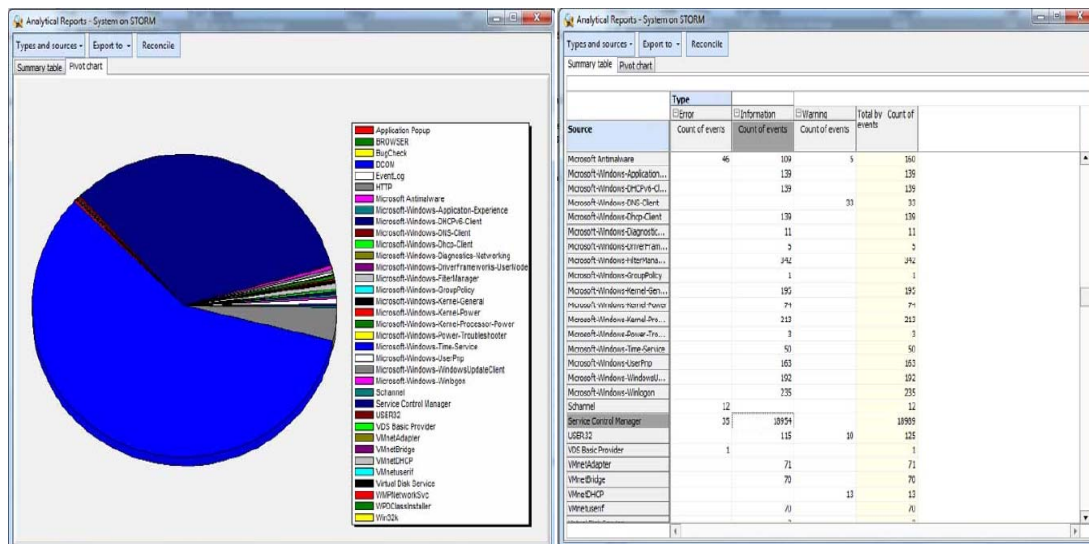


Ilustración 9: Event Log Explorer – Resultados Obtenidos
Fuente: eventlogxp [5]

- c) El usuario ha de poder elegir en qué formato (de entre al menos 2 posibles) quiere los informes resultantes del análisis ✓

Obtención de los informes en formatos distintos.

Event Log Explorer, a diferencia del Visor de Eventos de Windows, proporciona la posibilidad de imprimir uno o varios tipos de eventos, de manera conjunta o individual, pudiendo además exportarlos a otros formatos como HTML, Microsoft Excel o ficheros de texto delimitado por tabuladores, lo que permite la posterior interpretación y manipulación de los mismos por parte del usuario.

- d) La aplicación ha de permitir el envío de los resultados del análisis por e-mail ✗

Envío de los informes por correo electrónico

La aplicación no permite directamente el envío de los informes obtenidos por correo electrónico. Pero gracias a los formatos en los que se pueden exportar, el usuario podrá enviar los resultados por correo personalmente.

- e) Herramienta configurable por el usuario: El usuario ha de poder configurar los informes proporcionados: tipos de informe a obtener ✓, parámetros para el análisis ✗ y datos que componen los informes resultantes ✓

¿Qué puede hacer el usuario con los datos de entrada?

La función principal de Event Log Explorer es mostrar al usuario de forma amigable todos los eventos registrados en el fichero por el sistema, por lo que las opciones de configuración de la herramienta se dirigen principalmente a este servicio.

Event *Log Explorer* presenta una opción de carga de registro que le permite realizar un filtro de eventos previo en la etapa de carga del fichero, permitiendo al usuario cargar eventos de decenas de servidores Windows a la vez, pero sin que sea necesario cargar todos los eventos de todos los registros simultáneamente, sino sólo los requeridos en cada momento, teniendo también la opción de excluir y o cargar sólo aquellos tipos de eventos que le interese analizar.

Una vez filtrados y cargados los eventos que se quieren analizar, Event *Log Explorer* proporciona al usuario varias mini-funcionalidades para seleccionar los registros a mostrar, haciendo más fácil el seguimiento de los diferentes eventos y su posterior estudio al usuario:

- Proporciona un motor de búsqueda y filtrado de eventos muy potente, que permite al usuario final filtrar los eventos que aparecen en la lista por cualquier criterio que le interese, pudiendo incluso guardarse el resultado del filtrado para agilizar futuras consultas.
- Se puede configurar para hacer un seguimiento especial a un tipo de evento específico de manera que cada vez que se produzca un evento de este tipo, el usuario reciba una notificación directa que le ayude a obtener información de los problemas antes de que le afecten, dándole la oportunidad incluso de actuar de forma inmediata sobre los mismos.
- Además, permite codificar los identificadores de eventos por colores, haciéndole al usuario final más fácil la distinción entre los distintos tipos eventos analizados.

Las mini-funcionalidades que tiene Event *Log Explorer* nos facilitan una visión de la totalidad de los datos más reducida y específica, dándole al usuario la posibilidad de seleccionar sólo los datos que le interesen en cada momento, y mostrándole, únicamente, aquellos que puedan ser relevantes para su estudio; lo que le da una mayor facilidad a la hora de abstraer la información contenida en los mismos.

El análisis que obtenemos con esta herramienta, más que un análisis propiamente dicho, es un muestreo de los datos que tenemos, dejando al usuario final la tarea de abstraer la información contenida en ellos.

No permite modificar la lógica de análisis del sistema, sino, únicamente la configuración de los datos mostrados.

- f) **La aplicación debe contar con una interfaz de usuario que facilite la interacción con el sistema** ✓

Interfaz de Usuario

Event *Log Explorer* es un software orientado al cliente ya que la mayoría de las características avanzadas que presenta, han sido propuestas directamente por usuarios, lo que ha facilitado que su interfaz sea, en líneas generales, muy agradable e intuitiva de utilizar, permitiendo al usuario de la misma un control casi completo de los eventos que ocurren en el sistema y en las aplicaciones, y permitiéndole dar un paso más: desde la simple visualización de los datos de los eventos registrados, proporcionada por el sistema, a su análisis real.

Event *Log Explorer* proporciona dos tipos de interfaz de usuario diferentes en función de las preferencias del propio usuario: la interfaz de múltiples documentos (MDI) que permite poder abrir un número ilimitado de registros de eventos y colocarlos dentro de la misma ventana principal de Event *Log Explorer*; y la interfaz de documentos en pestañas (TDI) que permite abrir un número ilimitado de registros de eventos y que proporciona una mejor forma de navegar entre registros.

2.2.2 Performance Analysis of Logs (PAL)

Pal [6] es una herramienta indicada principalmente para analizar los contadores de rendimiento de un sistema.

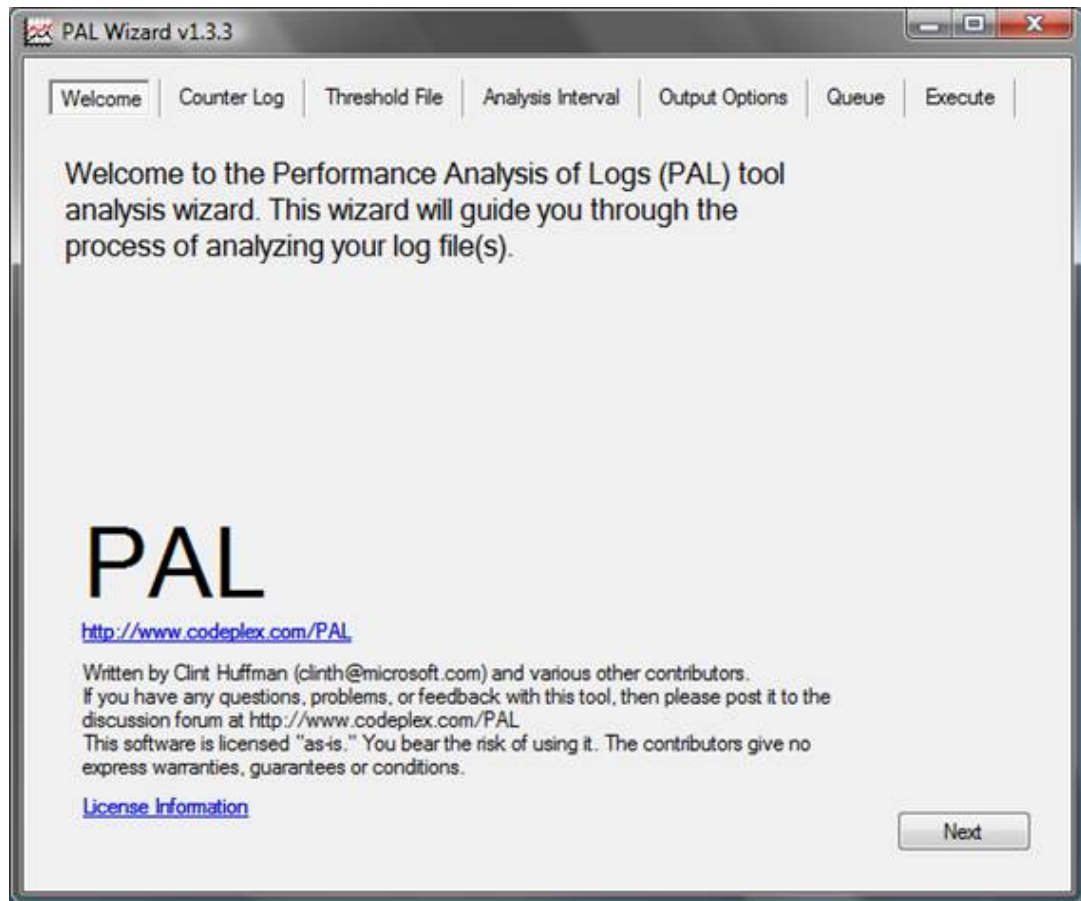


Ilustración 10: Performance Analysis of Logs (PAL) – Pantalla Principal.
Fuente: blogs.msdn [6]

RESUMEN DE LA APLICACIÓN

- Es una herramienta libre y de código abierto de Microsoft; creada especialmente para ayudar al usuario a detectar problemas de rendimiento en su ordenador.
- Como herramienta Microsoft que es, su función principal es la de analizar (bajo umbrales conocidos) los registros *log* de los contadores de rendimiento del sistema operativo Windows (o de alguna otra aplicación como Microsoft Exchange, SQL Server, SharePoint, Active Directory y Biztalk Server, entre otros) y proporcionar al usuario final reportes HTML con informes y análisis gráficos de los resultados obtenidos.
- PAL ha sido probado únicamente en Windows7, pero también debería funcionar en Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows Server 2008 R2 y Windows8.

CARACTERÍSTICAS BUSCADAS

a) La Aplicación ha de ser capaz de Analizar un Fichero Log ✓

¿Qué logs analiza? ¿Cuál es el formato de los datos de entrada?

PAL analiza los *logs* de los contadores de rendimiento del Sistema Operativo Windows y de la mayoría de los productos actuales de Microsoft como pueden ser SQL Server, Microsoft Exchange, ISS, Active Directory, MOSS, etc., e incluso algunos contadores diseñados por el propio usuario.

Puesto que su principal función es la de analizar los *logs* de los contadores de rendimiento, no es difícil entender que el formato de entrada aceptado por la aplicación sea .csv o .blg, y sólo funciona cuando los contadores están en inglés, por lo que si estamos trabajando con ficheros de *log* escritos en español, tendremos primero que traducirlos al inglés para que nuestra herramienta sea capaz de interpretar bien los nombres de los contadores.

b) La aplicación debe mostrar los informes obtenidos por pantalla ✓

Resultados mostrados por pantalla.

Pal genera reportes gráficos muy detallados del rendimiento de un servidor.

Una vez que PAL ha terminado de analizar el fichero .blg de entrada, se abre una ventana del navegador por defecto mostrando al usuario los resultados obtenidos del análisis realizado.

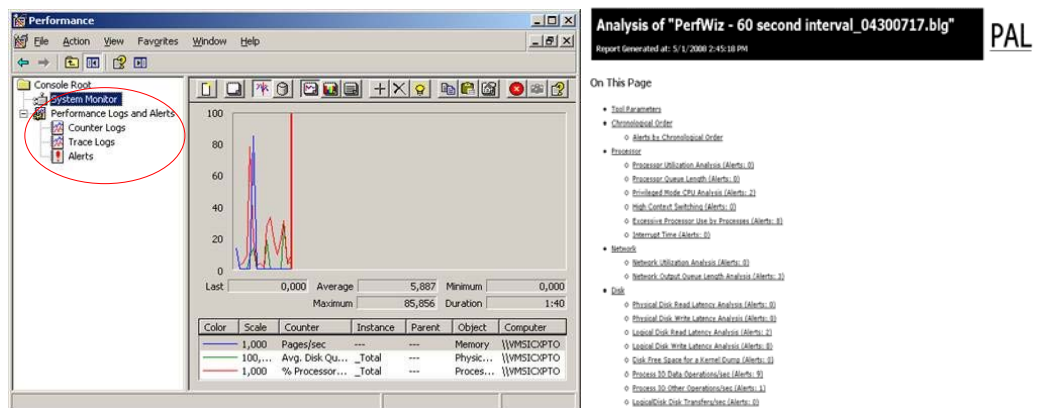


Ilustración 11: Performance Analysis of Logs (PAL) – Resultados Obtenidos.
Fuente: blogs.msdn [6] y blogs.technet [6]

- c) **El usuario ha de poder elegir en qué formato (de entre al menos 2 posibles) quiere los informes resultantes del análisis** ✓

Obtención de los informes en formatos distintos.

El reporte del análisis que genera PAL es un fichero HTML por defecto (aunque también nos da la posibilidad de exportar los resultados a .XML). Dicho fichero se guarda directamente en la carpeta de Mis Documentos\PAL Resultados y contiene hipervínculos y gráficos que permiten al usuario final una fácil interpretación y navegación por los resultados del análisis.

- d) **La aplicación ha de permitir el envío de los resultados del análisis por e-mail** ✗

Envío de los informes por correo electrónico

La aplicación no permite directamente el envío de los informes obtenidos por correo electrónico, y el envío personalmente del mismos, dado a los formatos en los que se pueden exportar los resultados, es posible, pero pesado.

- e) **Herramienta configurable por el usuario: El usuario ha de poder configurar los informes proporcionados: tipos de informe a obtener** ✓, **parámetros para el análisis** ✓ **y datos que componen los informes resultantes** ✓

¿Qué puede hacer el usuario con los datos de entrada?

PAL no es un intento de remplazar el análisis tradicional del rendimiento de un sistema, pero sí que es una herramienta muy útil para agilizar el tiempo de análisis del mismo.

En líneas generales su uso es muy rápido e intuitivo, pues las opciones que vienen por defecto marcadas en la aplicación suelen ser más que suficientes para un correcto análisis del rendimiento del sistema. No obstante, también permite al usuario elegir varios aspectos del análisis que puede hacer que resulte mucho más exhaustivo, pero lento; como por ejemplo, el intervalo de análisis a tener en cuenta para analizar el fichero de entrada - cuantos más grupos de análisis se hagan, más tiempo tardará la aplicación en realizar dicho análisis, pero más fehacientes serán los resultados obtenidos.

Una opción muy a tener en cuenta a la hora de empezar a trabajar con el fichero de entrada, es que la aplicación permite analizarlo bajo diferentes ficheros de umbrales conocidos, de manera que es muy importante escoger bien el que mejor encaje con las necesidades requeridas.

Además, PAL también permite modificar dicho fichero de umbrales conocidos pudiendo añadir y/o editar tanto los contadores especificados en el mismo como el valor de los umbrales definidos por defecto para ellos, pudiendo incluso concretar nuevos umbrales que pudieran servir de ayuda para el análisis a realizar.

PAL también permite encolar varios análisis y ejecutarlos todos seguidos de una vez, lo que permite abarcar un mayor número de aspectos para el análisis.

También proporciona una serie de filtros que permiten al usuario poder especificar el contador o contadores que quiere analizar en cada momento. Esto, teniendo en cuenta que hay diversidad de contadores de rendimiento en un ordenador, y que sus valores varían constantemente, es una gran ventaja, ya que da la posibilidad al usuario de no tener que ver todos y cada uno de los valores anotados por cada contador, permitiéndole seleccionar únicamente aquellos que le interese estudiar en cada momento, de manera que la aplicación sólo le muestre las oscilaciones y variaciones del mismo durante el período de tiempo que se haya determinado.

Los reportes que nos proporciona PAL son muy completos y fáciles de seguir, ya que por cada contador de rendimiento seleccionado para analizar, la aplicación muestra una breve descripción del mismo, una gráfica con los valores obtenidos, y una tabla resumen con los umbrales alcanzados.

Sí que permite modificar la lógica de análisis del sistema, pudiéndose configurar los umbrales de cada contador y diferentes parámetros de los mismos para la realización de los análisis.

- f) La aplicación debe contar con una interfaz de usuario que facilite la interacción con el sistema ✓**

Interfaz de Usuario

PAL presenta una interfaz básica que guía al usuario por los pasos necesarios para empezar el proceso de análisis sin mayor complicación, permitiéndole interactuar con él para pedirle, de forma sencilla, toda la información específica del sistema que necesita para analizar los contadores de rendimiento de la manera más precisa posible. Dicha información suelen ser variables que afectan directa o indirectamente al rendimiento del sistema en sí: total de memoria física instalada, número de procesadores del sistema, si el sistema es 64-bit o 32-bit, etc.; pero además, también permite al diseñador de la aplicación realizar preguntas directas al usuario final que ayuden a establecer, más concretamente, los umbrales de los contadores que se van a analizar, así como el tipo de análisis que quiere realizar.

2.2.3 Log Parser

Log Parser [7] se ejecuta principalmente desde una interfaz de línea de comandos que en verdad no es muy intuitiva, pero tiene también una interfaz de ejecución más amigable e intuitiva.



```
C:\Program Files\Log Parser 2.2>logparser /h

Microsoft (R) Log Parser Version 2.2.10
Copyright (C) 2004 Microsoft Corporation. All rights reserved.

Usage:  LogParser [-i:<input_format>] [-o:<output_format>] <SQL query> ;
        file:<query_filename>[?param1=value1+...]
        [<input_format_options>] [<output_format_options>]
        [-q[:ON|OFF]] [-e:<max_errors>] [-iw[:ON|OFF]]
        [-stats[:ON|OFF]] [-saveDefaults] [-queryInfo]

        LogParser -c -i:<input_format> -o:<output_format> <from_entity>
        <into_entity> [<where_clause>] [<input_format_options>]
        [<output_format_options>] [-multiSite[:ON|OFF]]
        [-q[:ON|OFF]] [-e:<max_errors>] [-iw[:ON|OFF]]
        [-stats[:ON|OFF]] [-queryInfo]

-i:<input_format>      : one of IISW3C, NCSA, IIS, IISODBC, BIN, IISMSID,
                        HTTPERR, URLSCAN, CSU, TSU, W3C, XML, EUT, ETW,
                        NETMON, REG, ADS, TEXTLINE, TEXTWORD, FS, COM (if
                        omitted, will guess from the FROM clause)
-o:<output_format>    : one of CSU, TSU, XML, DATAGRID, CHART, SYSLOG,
                        NEUROVIEW, NAT, W3C, IIS, SQL, TPL, NULL (if omitted,
                        will guess from the INTO clause)
-q[:ON|OFF]          : quiet mode; default is OFF
-e:<max_errors>       : max # of parse errors before aborting; default is -1
                        (ignore all)
-iw[:ON|OFF]         : ignore warnings; default is OFF
-stats[:ON|OFF]      : display statistics after executing query; default is
                        ON
-c                   : use built-in conversion query
-multiSite[:ON|OFF]  : send BIN conversion output to multiple files
                        depending on the SiteID value; default is OFF
-saveDefaults        : save specified options as default values
-restoreDefaults     : restore factory defaults
-queryInfo           : display query processing information (does not
                        execute the query)

Examples:
LogParser "SELECT date, REVERSEDNS(c-ip) AS Client, COUNT(*) FROM file.log
WHERE sc-status<>200 GROUP BY date, Client" -e:10
LogParser file:myQuery.sql?myInput=C:\temp\ex*.log+myOutput=results.csv
LogParser -c -i:BIN -o:W3C file1.log file2.log "ComputerName IS NOT NULL"

Help:
-h GRAMMAR            : SQL Language Grammar
-h FUNCTIONS [ <function> ] : Functions Syntax
-h EXAMPLES          : Example queries and commands
-h -i:<input_format>    : Help on <input_format>
-h -o:<output_format>   : Help on <output_format>
-h -c                 : Conversion help

C:\Program Files\Log Parser 2.2>logparser "Select count(distinct c-ip) from ex07
0102.log"
COUNT(DISTINCT c-ip)
-----
2177

Statistics:
-----
Elements processed: 6384
Elements output:    1
Execution time:     0.03 seconds

C:\Program Files\Log Parser 2.2>
```

Ilustración 12: Log Parser – Pantalla Principal
Fuente: wikimedia [7]

RESUMEN DE LA APLICACIÓN

- Es una herramienta gratuita creada por Microsoft, increíblemente útil y versátil, utilizada principalmente para obtener la información contenida en ficheros *log*.
- Está formada por tres componentes básicos: un motor de entrada, un motor de consultas SQL y un motor de salida. Utiliza un lenguaje similar al SQL genérico y acepta prácticamente todos los formatos de entrada más comunes como pueden ser los registros proporcionados por el IIS (Internet Information Server), las entradas del visor de eventos de Windows, los ficheros XML, los ficheros CSV y, el sistema de archivos, entre otros. Además, permite exportar los resultados de las consultas realizadas a distintos tipos de formatos según tu elección: CSV, XML, SQL Server, gráficos, etc.
- Es una herramienta de Microsoft y por tanto se puede utilizar en todas las versiones de Windows 2000, Windows ME, Windows XP, Windows Vista, Windows 7, Windows 8 y funciona perfectamente con IIS 5, 6, 7 y 7.5.

CARACTERÍSTICAS BUSCADAS

a) La Aplicación ha de ser capaz de Analizar un Fichero *Log* ✓

¿Qué *logs* analiza? ¿Cuál es el formato de los datos de entrada?

Log Parser es una herramienta capaz de analizar una gran variedad de tipos de registros entre los que destacan principalmente los ficheros *log* de IIS (Internet Information Server) y otros archivos de registros propios de Microsoft Windows® como: el visor de eventos, o el propio sistema de archivos.

De aquí que los formatos de ficheros de entrada que admite, sean también muy diversos, en función del tipo de registro que se quiera analizar.

FORMATO DE ENTRADA	TIPO DE REGISTRO
W3C	<i>Logs</i> de IIS
HTTERR	<i>Logs</i> de HTTP.sys
EVT	<i>Logs</i> de eventos NT
ETW	<i>Logs</i> de ETW
URLSCAN	<i>Logs</i> de URLSCAN
NETMON	Trazas de Network Monitor
XML	Ficheros XML genéricos
TEXTWORD	Ficheros TXT genéricos

Tabla 2: *Log Parser* – Ejemplos de Formatos de Entrada admitidos
Fuente: technet.microsoft y blogs.msdn [7]

Esto es posible gracias a que *Log Parser* cuenta con un formato personalizable “.COM” que le permite crear su propio formato de entrada, totalmente personalizado, y conectarlo al motor de entrada del sistema.

b) La aplicación debe mostrar los informes obtenidos por pantalla ✓

Resultados mostrados por pantalla.

Log Parser ser una herramienta impresionante, que permite obtener los resultados de las consultas realizadas en prácticamente todos los tipos de formatos que podamos imaginar, sólo con indicárselo en la línea de ejecución de la consulta.

Log Parser te permite, entre otras cosas:

- Visualizar una consulta de datos por pantalla directamente:

```

Administrator: Log Parser 2.2
C:\Program Files\Log Parser 2.2>logparser.exe -i:EVT "SELECT TOP 5 TimeGenerated
, EventID FROM System WHERE EventID>10"
TimeGenerated      EventID
-----
2012-07-15 10:24:08 6011
2012-07-15 10:24:08 6009
2012-07-15 10:24:08 6005
2012-07-15 10:24:09 263
2012-07-15 10:23:16 4201

Statistics:
-----
Elements processed: 5
Elements output:    5
Execution time:     0.01 seconds

C:\Program Files\Log Parser 2.2>
  
```

Path	Name	Size	Attributes
C:\Program Files\Microsoft Office\Office10\WINWORD.EXE	WINWORD.EXE	10738448	-A--R----
C:\Program Files\Microsoft Office\Office10\EXCEL.EXE	EXCEL.EXE	9358096	-A--R----
C:\Program Files\Microsoft Office\Office10\OUTLLIB.DLL	OUTLLIB.DLL	6326984	-A-----
C:\Program Files\Microsoft Office\Office10\POWERPNT.EXE	POWERPNT.EXE	6093584	-A--R----
C:\Program Files\Microsoft Office\Office10\MSOWC.DLL	MSOWC.DLL	3041880	-A-----
C:\Program Files\Microsoft Office\Office10\CLIPPIT.ACS	CLIPPIT.ACS	2904417	-A-----
C:\Program Files\Microsoft Office\Office10\GRAPH.EXE	GRAPH.EXE	2144824	-A-----
C:\Program Files\Microsoft Office\Office10\1033\OUTLLIBR.DLL	OUTLLIBR.DLL	1977032	-A-----
C:\Program Files\Microsoft Office\Office10\1033\XLMAIN10.CHM	XLMAIN10.CHM	1646072	-A-----
C:\Program Files\Microsoft Office\Office10\MSOWCW.DLL	MSOWCW.DLL	1200177	-A-----

Estadísticas			
Elements processed:	1000		
Elements output:	10		
Execution time:	0.42 seconds		

Ilustración 13: *Log Parser* – Resultados Obtenidos por pantalla
Fuente: [blog.extreme-advice](http://blog.extreme-advice.com) y [support.microsoft](http://support.microsoft.com) [7]

- Visualizar el resultados de una consulta por pantalla en un “datagrid” (tabla):

```
logparser.exe -i:EVT -o:DATAGRID
```

```
"SELECT * FROM System WHERE EventID>10"
```

EventL...	RecordNum...	TimeGenerated	TimeWitten	Event...	EventTy...	EventTypeNa...	EventCateg...	EventCategoryNa...	SourceName	Stri...	ComputerNa...	SID	Message	Data
System	1	2012-07-15 10:24...	2012-07-15 10:24...	6011	4	Information ev...	0	None	EventLog	26L...	26L2233A5-07	NULL	The NetBIOS name and DNS ...	NULL
System	2	2012-07-15 10:24...	2012-07-15 10:24...	6009	4	Information ev...	0	None	EventLog	6.0...	26L2233A5-07	NULL	Microsoft (R) Windows (R) 6.0...	NULL
System	3	2012-07-15 10:24...	2012-07-15 10:24...	6005	4	Information ev...	0	None	EventLog	NULL	26L2233A5-07	NULL	The Event log service was sta...	DC0707000...
System	4	2012-07-15 10:24...	2012-07-15 10:24...	263	2	Warning event	0	None	PlugPlayMana...	She...	26L2233A5-07	NULL	The service 'ShellHWDetectio...	NULL
System	5	2012-07-15 10:23...	2012-07-15 10:23...	4201	4	Information ev...	0	None	Tcpip	ILoo...	26L2233A5-07	NULL	The system detected that net...	00000000020...
System	6	2012-07-15 10:23...	2012-07-15 10:23...	4201	4	Information ev...	0	None	Tcpip	ILoo...	26L2233A5-07	NULL	The system detected that net...	00000000020...
System	14	2012-07-15 10:24...	2012-07-15 10:24...	134	2	Warning event	0	None	W32Time	No ...	26L2233A5-07	NULL	NtpClient was unable to set a ...	NULL
System	15	2012-07-15 10:24...	2012-07-15 10:24...	15016	1	Error event	0	None	HTTP	\De...	26L2233A5-07	NULL	The description for Event ID 1...	00000400020...
System	16	2012-07-15 10:24...	2012-07-15 10:24...	134	2	Warning event	0	None	W32Time	No ...	26L2233A5-07	NULL	NtpClient was unable to set a ...	NULL
System	17	2012-07-15 10:24...	2012-07-15 10:24...	1008	4	Information ev...	0	None	Dhcpv6	%2	26L2233A5-07	NULL	Your computer was unable to i...	02000000

Line Num...	Date	Time	Client...	Server...	Server P...	Request V...	Request URI	Bytes s...	Stat...	Sub-stat...	Win 32 Stat...	Time Tak...
17	2006-08...	12:13...	127...	127.0...	80	GET	/Configuratio...	NULL	302	0	0	16359
36	2006-08...	12:14...	127...	127.0...	80	GET	/Parent/Web...	NULL	500	0	0	1609
38	2006-08...	12:14...	127...	127.0...	80	GET	/Sukesh/Def...	NULL	200	0	0	890
47	2006-08...	12:14...	127...	127.0...	80	GET	/AbhinavSam...	NULL	200	0	0	750
44	2006-08...	12:14...	127...	127.0...	80	GET	/TransmitFile/...	NULL	200	0	0	453
15	2006-08...	12:13...	127...	127.0...	80	GET	/Configuratio...	NULL	401	1	0	125
18	2006-08...	12:13...	127...	127.0...	80	GET	/Configuratio...	NULL	200	0	0	109
30	2006-08...	12:14...	127...	127.0...	80	GET	/Parent	NULL	401	1	0	109
22	2006-08...	12:13...	127...	127.0...	80	GET	/Configuratio...	NULL	200	0	0	78

Ilustración 14: Log Parser – Resultados Obtenidos en una tabla

Fuente: blog.extreme-advice y support.microsoft [7]

- Visualizar el resultados de una consulta en un gráfico 3D:

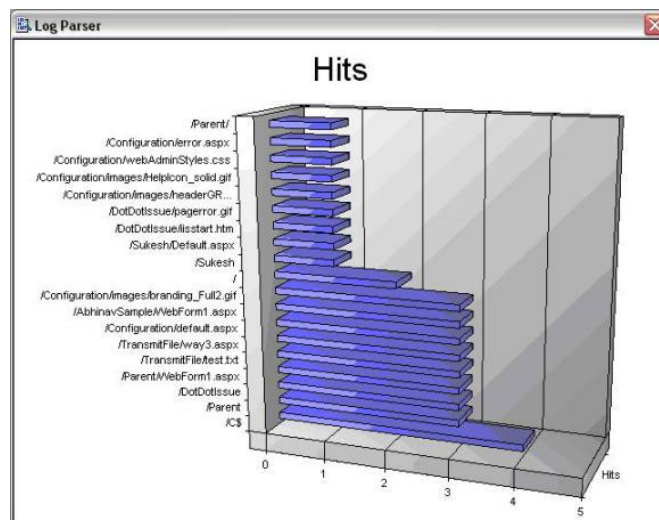


Ilustración 15: Log Parser – Resultados Obtenidos en un Gráfico de Barras 3D

Fuente: support.microsoft [7]

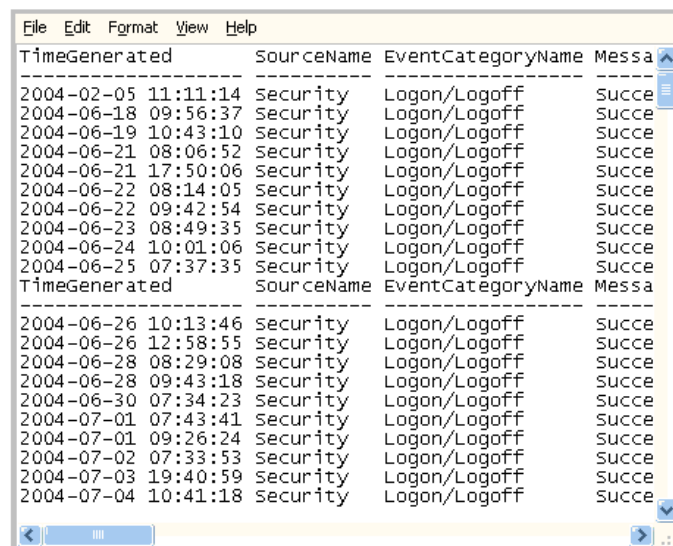
- Buscar un tipo de *log* específico:

```
C:\>LogParser

"SELECT
    TimeGenerated, SourceName, EventCategoryName, Message
INTO report.txt
FROM Security
WHERE EventID = 528
    AND SID LIKE '%TESTUSER%' "

-resolveSIDs:ON
```

Ilustración 16: Log Parser – Ejemplo de Consulta I
Fuente: technet.microsoft [7]



TimeGenerated	SourceName	EventCategoryName	Message
2004-02-05 11:11:14	Security	Logon/Logoff	Success
2004-06-18 09:56:37	Security	Logon/Logoff	Success
2004-06-19 10:43:10	Security	Logon/Logoff	Success
2004-06-21 08:06:52	Security	Logon/Logoff	Success
2004-06-21 17:50:06	Security	Logon/Logoff	Success
2004-06-22 08:14:05	Security	Logon/Logoff	Success
2004-06-22 09:42:54	Security	Logon/Logoff	Success
2004-06-23 08:49:35	Security	Logon/Logoff	Success
2004-06-24 10:01:06	Security	Logon/Logoff	Success
2004-06-25 07:37:35	Security	Logon/Logoff	Success
2004-06-26 10:13:46	Security	Logon/Logoff	Success
2004-06-26 12:58:55	Security	Logon/Logoff	Success
2004-06-28 08:29:08	Security	Logon/Logoff	Success
2004-06-28 09:43:18	Security	Logon/Logoff	Success
2004-06-30 07:34:23	Security	Logon/Logoff	Success
2004-07-01 07:43:41	Security	Logon/Logoff	Success
2004-07-01 09:26:24	Security	Logon/Logoff	Success
2004-07-02 07:33:53	Security	Logon/Logoff	Success
2004-07-03 19:40:59	Security	Logon/Logoff	Success
2004-07-04 10:41:18	Security	Logon/Logoff	Success

Ilustración 17: Log Parser – Resultado Mostrado I
Fuente: technet.microsoft [7]

- Calcular Estadísticas y visualizar los resultados en un gráfico:

```
C:\>LogParser

"SELECT sc-status, COUNT(*) AS Times INTO Chart.gif
FROM <1>
GROUP BY sc-status
ORDER BY Times DESC"

-chartType:PieExploded3D
-chartTitle:"Status Codes"
```

Ilustración 18: Log Parser – Ejemplo de Consulta II
Fuente: technet.microsoft [7]

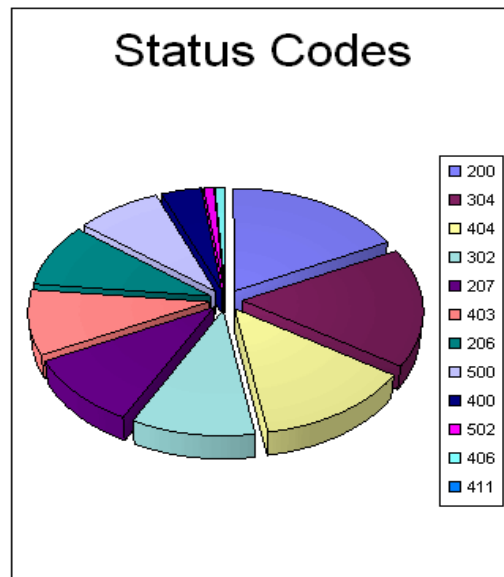


Ilustración 19: Log Parser – Resultado Mostrado II
Fuente: technet.microsoft [7]

- c) El usuario ha de poder elegir en qué formato (de entre al menos 2 posibles) quiere los informes resultantes del análisis ✓

Obtención de los informes en formatos distintos.

Tal y como se ha comentado anteriormente, *Log Parser* es un ligero motor de búsqueda basado en SQL que es capaz de operar en un asombroso número de formatos de entrada diferente; proporcionando, igualmente, un asombroso número de formatos de salida distinto:

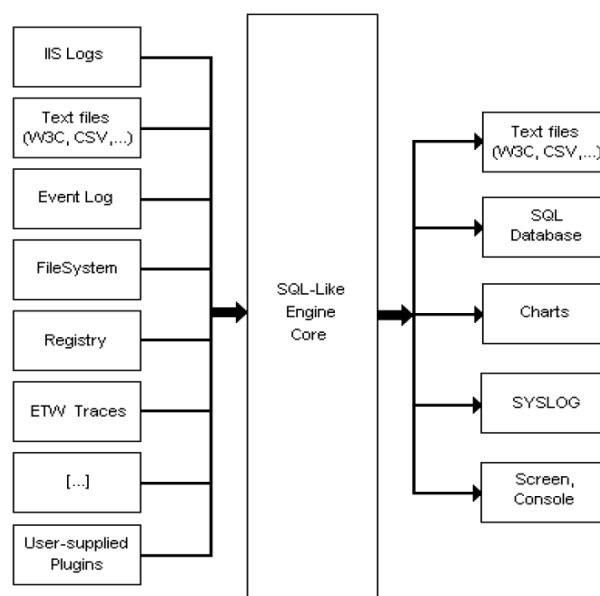


Ilustración 20: Log Parser - Diagrama de formatos de entrada y salida
Fuente: josemanuelmartin [7]

El formato de los resultados de su consulta se puede personalizar en un resultado basado en texto o bien, se pueden traspasar a destinos más especializados como SQL, SYSLOG o un gráfico, entre otros.

Formato Salida	Descripción
CHART	Crea archivos de imágenes que contienen gráficos de los valores de campo de los registros de salida (requiere Microsoft Office 2000 o posterior)
CSV	Escribe los registros de salida como texto de valores delimitados por comas. (Comma Separated Value)
DATAGRID	Muestra los registros de salida en una interfaz gráfica de usuario.
IIS	Escribe los registros de salida en el formato de archivo de registro de Microsoft IIS.
NAT	Escribe los registros de salida en un formato legible de columnas tabuladas.
SQL	Carga los registros de salida en una tabla de una base de datos SQL.
SYSLOG	Envía mensajes a un servidor Syslog o a un archivo de texto con formato SYSLOG ; crea archivos de texto que contienen mensajes Syslog; envía mensajes Syslog a los usuarios.
TPL	Escribe los registros de salida con un formato acorde con las plantillas definidas por el usuario.
TSV	Escribe los registros de salida como texto de valores delimitados por tabuladores o por espacios. (Tab Separated Value)
W3C	Escribe los registros de salida en el formato de archivo de registro extendido W3C.
XML	Escribe los registros de salida como nodos de documentos XML.

Tabla 3: Log Parser – Ejemplos de Formatos de Salida
Fuente: technet.microsoft [7]

En el punto anterior, hemos podido ver varios ejemplos con distintos de estos formatos de salida posibles.

d) La aplicación ha de permitir el envío de los resultados del análisis por e-mail ✖

Envío de los informes por correo electrónico

Gracias a todos los formatos de salida que proporciona la herramienta, si bien no es una opción de la propia aplicación el enviar directamente los resultados obtenidos por e-mail, sí que es permite al usuario poder enviarlos después personalmente por correo electrónico (aunque esto sólo será posible, si el usuario ha seleccionado un formato de salida del informe compatible con esta función).

- e) **Herramienta configurable por el usuario:** El usuario ha de poder configurar los informes proporcionados: tipos de informe a obtener ✓, parámetros para el análisis ✗ y datos que componen los informes resultantes ✓

¿Qué puede hacer el usuario con los datos de entrada?

La mayor parte de los programas, generalmente se diseñan para alcanzar un número limitado de tareas específicas. Pero, *Log Parser* es diferente...

La cantidad de maneras en que se puede usar ésta herramienta, sólo están limitadas por las necesidades y la imaginación del usuario.

Usted le dice a *Log Parser* qué información necesita y cómo desea procesarla, y él lo hace todo.

Log Parser es una herramienta muy versátil y eficaz, ya que gracias a que incorpora la tecnología de SQL, puede analizar gran cantidad de datos de forma rápida y eficiente y buscar información, en varios gigabytes de archivos de diferentes tipos de registro, para obtener los datos realmente necesarios y relevantes, al mismo tiempo que filtra y descarta el ruido.

Con la última versión de *Log Parser* se puede hacer cosas como:

- Buscar en los registros de sucesos y localizar y mostrar rápidamente sucesos de interés.
- Buscar en el sistema de archivos y localizar y mostrar rápidamente archivos y carpetas de interés.
- Buscar en Active Directory y localizar y mostrar rápidamente objetos de interés.
- Buscar en los registros de Servicios de Internet Information Server (IIS), en los archivos de captura de Netmon, en el Registry y en todo tipo de ficheros de *logs*, rápida y fácilmente.
- Buscar archivos de registro de cualquier tipo y formato.

Pero eso no es ni la mitad de lo que puede hacer esta herramienta ya que además, gracias al lenguaje de consultas SQL que utiliza, el usuario puede agregar todas las consultas que considere necesarias para manipular mejor los datos que quiere analizar.

Si bien es cierto que con las consultas SQL que proporciona se pueden obtener infinidad de combinaciones de los datos mostrados, no permite dar un peso/valor a los distintos parámetros analizados, por lo que podemos decir que no admite la modificación y codificación de la lógica de análisis del sistema según las necesidades del usuario, sino, únicamente la configuración de los informes obtenidos.

- f) La aplicación debe contar con una interfaz de usuario que facilite la interacción con el sistema ✖

Interfaz de Usuario

Log Parser no presenta una interfaz de usuario como tal. Su uso es principalmente realizado siempre desde la línea de comandos y, para que podamos entender bien por qué decimos esto, vamos a exponer a continuación un ejemplo de ejecución de una consulta propia con esta herramienta.

En general, las secuencias de comandos de *Log Parser* constan de cinco pasos principales para su ejecución:

1. Crear una instancia del objeto *Log Parser*

```
Set objLogParser = CreateObject("MSUtil.LogQuery")
```

2. Crear y configurar el objeto *InputFormat* para especificar de dónde se obtienen los datos y el formato del fichero y los registros de entrada.

(Para el fichero de *logs* del sistema de archivos)

```
Set objInputFormat = _  
    CreateObject("MSUtil.LogQuery.FileSystemInputFormat")
```

En función del tipo de formato de entrada que escojamos, habrá una serie de propiedades opcionales que podremos configurar (también por línea de comandos).

3. Crear y configurar el objeto *OutputFormat* para especificar el formato de salida escogido: XML, gráficos, CSV, etc.

```
Set objOutputFormat = _  
    CreateObject ("MSUtil.LogQuery.NativeOutputFormat")
```

Al igual que con el formato de entrada, en función del tipo de formato de salida que escojamos, habrá una serie de propiedades opcionales que podremos configurar (también por línea de comandos).

4. Crear una consulta de Log Parser con la tecnología SQL

En el manual de Ayuda de Microsoft Log Parser, podremos encontrar prácticamente todos los campos disponibles para los todos los formatos de entrada que es capaz de leer esta herramienta.

(Propiedades a analizar *Name* y *Size*)

```
strQuery =  
"SELECT Name, Size FROM 'C:\Scripts\*.*' ORDER BY Name ASC"
```

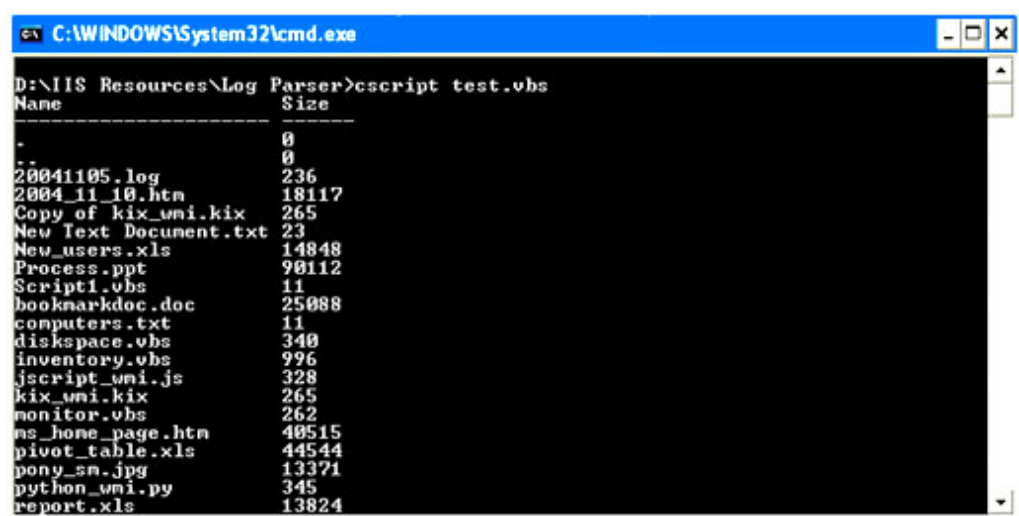
5. Ejecutar la consulta y devolver los datos resultantes

Para ello, se llama al método ExecuteBatch y se le pasan los tres parámetros configurados anteriormente: la variable que contiene la consulta SQL (strQuery); la referencia al objeto que especifica el formato de entrada InputFormat (objInputFormat); y la referencia al objeto que representa el formato de salida (objOutputFormat).

```
objLogParser.ExecuteBatch strQuery, objInputFormat, objOutputFormat
```

El método ExecuteBatch ejecuta una secuencia de comandos y, luego, muestra o guarda los resultados en el formato elegido: una ventana de comandos, un archivo HTML, una base de datos SQL, etc.

El formato de salida nativa seleccionado, muestra los datos en un formato tabular en la ventana de comandos:



Name	Size
..	0
20041105.log	236
2004_11_10.htm	18117
Copy of kix_wmi.kix	265
New Text Document.txt	23
New_users.xls	14848
Process.ppt	90112
Script1.vbs	11
bookmarkdoc.doc	25088
computers.txt	11
diskspace.vbs	340
inventory.vbs	996
jscrip_t_wmi.js	328
kix_wmi.kix	265
monitor.vbs	262
ns_hone_page.htm	40515
pivot_table.xls	44544
pony_sn.jpg	13371
python_wmi.py	345
report.xls	13824

Ilustración 21: Log Parser – Resultados Obtenidos

Log Parser, también permite realizar los cinco pasos de su ejecución, en una sola línea:

```
C:\program files\log parser 2.2> logparser.exe strQuery -i:[InputFormat]
-o:[ OutputFormat] -rtp:-1
```

No obstante, a pesar de no tener una interfaz de usuario propiamente dicha (GUI) que funcione como *front-end* de la herramienta, sí que existen actualmente en el mercado aplicaciones que complementan a la perfección este detalle.

Una de ellas, es *Log Parser Lizard* [8] (*Microsoft Log Parser GUI*) que ha sido creada principalmente para complementar la interfaz en línea de comandos de *log parser* (ya que si no se está familiarizado con ella, si bien es muy potente y rápida, puede resultar poco intuitiva y ágil para el usuario).

Esta herramienta permite, entre otras cosas, el manejo de *queries* para la elaboración de los informes con *Log Parser* y la exportación de los resultados obtenidos a Excel, gráficos, etc. de una forma mucho más intuitiva y amigable.

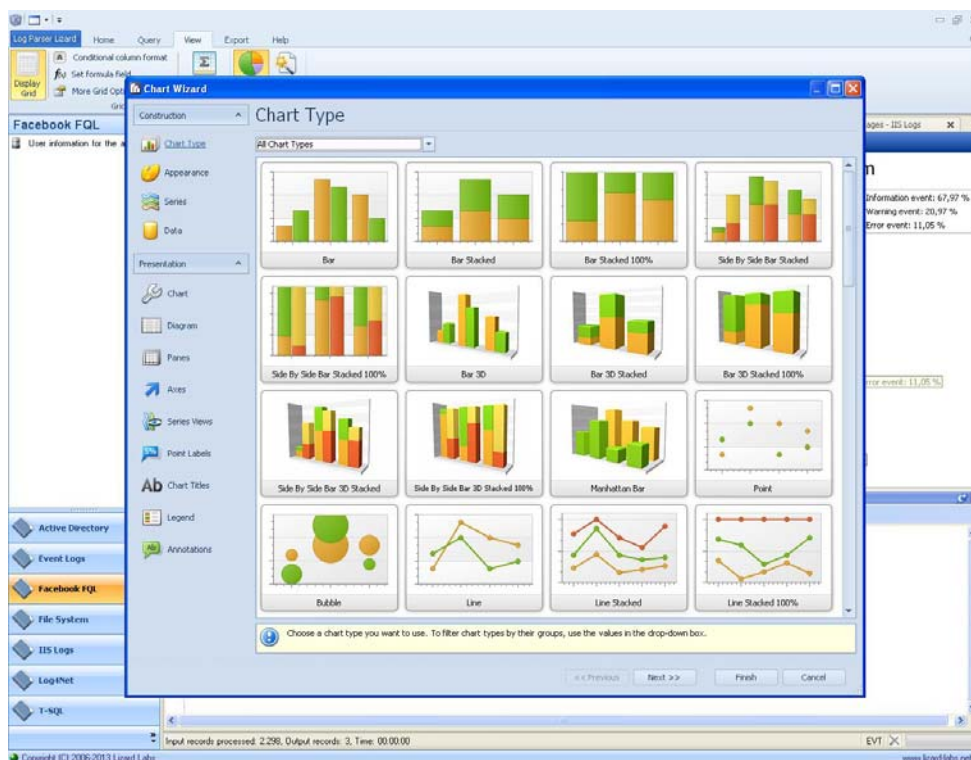


Ilustración 22: Microsoft Log Parser GUI - Log Parser Lizard

Otra herramienta de este tipo podría ser *Visual Log Parser* [9] y otra, un poco más completa, *Log Parser Studio* [10].

2.2.4 Analog

Una característica particular de Analog [11] es que, a diferencia de otros programas, no contiene una interfaz gráfica de usuario (o GUI) si no que consiste en un archivo ejecutable (analog.exe) que se lanza desde la línea de comandos y al que se le pasan una serie de parámetros para configurarlo y poder crear los reportes y listados con la información analizada.

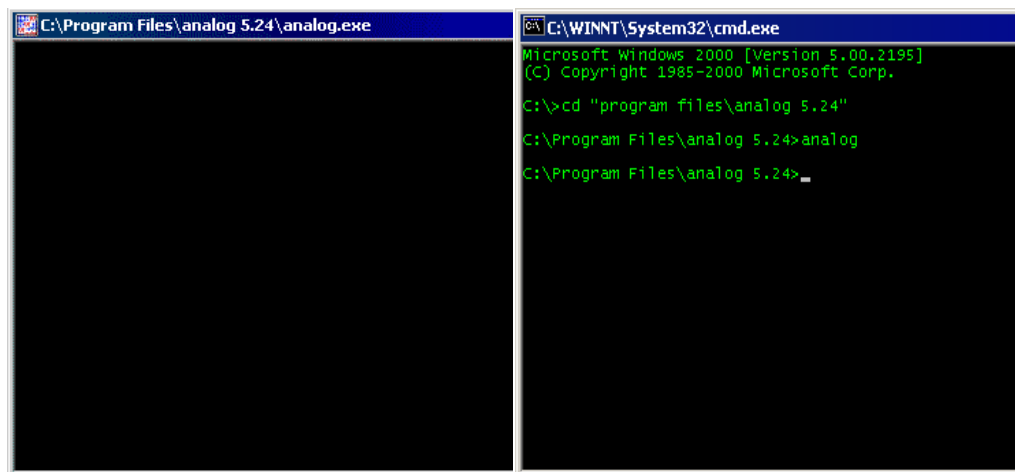


Ilustración 23: Analog - Arranque y Lanzamiento de la aplicación

RESUMEN DE LA APLICACIÓN

- a. Es un programa de ordenador gratuito, de licencia libre y código abierto (desde 2004) utilizado principalmente para medir el uso de un servidor web.
- b. Es un analizador simple, pero bastante completo, creado especialmente analizar ficheros *log* de servidores web, proporcionando la posibilidad de hacer búsquedas DNS inversas en los ficheros *log* del servidor para averiguar de dónde provienen las llamadas. Además, puede funcionar en host virtuales, es decir para múltiples dominios creados en un mismo servidor, por lo que se suele decir que es compatible con archivos de registro de múltiples máquinas virtuales. También puede analizar varios tipos diferentes de *logs* de servidores web incluyendo Apache, IIS, e iPlanet, y proporciona al usuario más de 200 opciones de configuración diferentes, pudiendo generar 32 reportes de análisis distintos, en 35 idiomas posibles.
- c. Está escrito en C estándar, por lo que debería compilar prácticamente en todas las máquinas que tengan un compilador de C. De hecho, que se sepa, funciona en todos los entornos conocidos: Windows, Mac OS, Linux y demás sistemas operativos tipo UNIX. Además, actualmente también funciona en MSIE, iOS, Android, Windows Phone y Windows 8/Server 2012 R2.

CARACTERÍSTICAS BUSCADAS

a) La Aplicación ha de ser capaz de Analizar un Fichero Log ✓

¿Qué logs analiza? ¿Cuál es el formato de los datos de entrada?

Analog es una herramienta creada principalmente para analizar los ficheros de *log* proporcionados por diferentes tipos de servidores web conocidos entre lo que destacan Apache e IIS (Internet Information Server).

Puede leer prácticamente casi todos los formatos de ficheros *log* conocidos, siendo capaz además, de analizar formatos diseñados específicamente por el usuario.

Normalmente trabaja con el formato común, el formato combinado NCSA, y los formatos de Netscape, WebStar y Netpresenz.

Pero además, también es capaz de analizar formatos diseñados específicamente por el usuario.

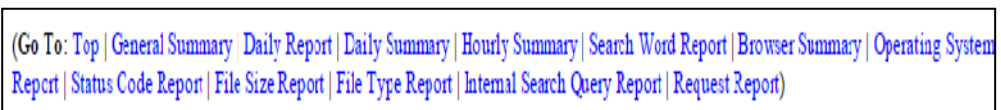
b) La aplicación debe mostrar los informes obtenidos por pantalla ✓

Resultados mostrados por pantalla.

El reporte de resultados que presenta Analog después del análisis, es bastante completo y está muy bien organizando, lo que permite al usuario un mejor análisis del mismo.

La capacidad gráfica de esta herramienta no es de las mejores, de hecho, se suele decir que no tiene un gran potencial generando gráficas y que sus reportes son bastante sencillos, pero esto no debería de considerarse un problema, ya que mejora en cada nueva versión que sale e incluso existen ya paquetes adicionales de la herramienta que añaden nuevas posibilidades para exportar los resultados obtenidos a otros programas específicos de gráficos que completen los informes.

Al principio del todo, y después de cada punto del análisis, el resumen presentado al usuario incluye siempre un pequeño índice completo con enlaces a todos los puntos analizados, para facilitar la navegación por todos los informes presentados.



(Go To: [Top](#) | [General Summary](#) | [Daily Report](#) | [Daily Summary](#) | [Hourly Summary](#) | [Search Word Report](#) | [Browser Summary](#) | [Operating System Report](#) | [Status Code Report](#) | [File Size Report](#) | [File Type Report](#) | [Internal Search Query Report](#) | [Request Report](#))

Ilustración 24: Analog - Índice para la navegación en un Informe

Como podemos observar en los apartados del índice del ejemplo expuesto el análisis de Analog puede incluir diversos reportes diferentes como, el reporte de estadísticas generales; el reporte diario; el resumen de las peticiones hechas por día o por horas; información sobre las peticiones de búsqueda más utilizadas para llegar al sitio web; información sobre los navegadores de los usuarios, etc.

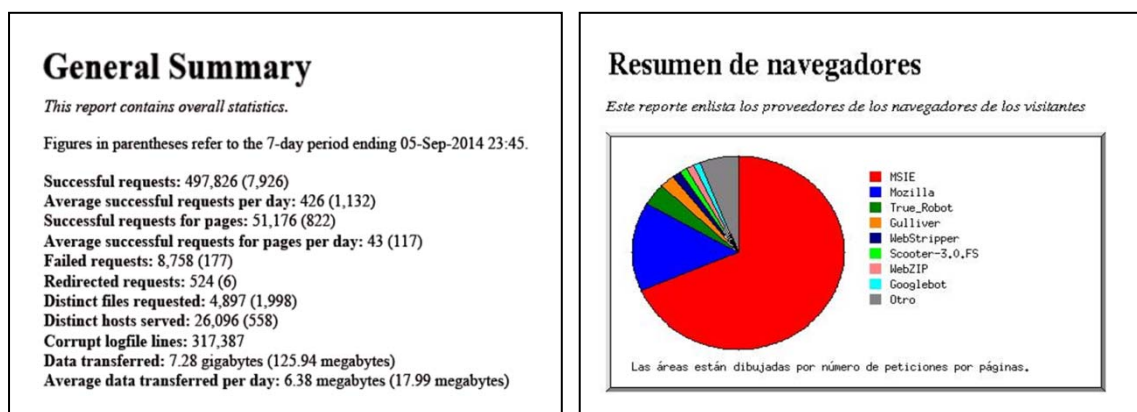


Ilustración 25: Analog – Resultados Obtenidos

- c) El usuario ha de poder elegir en qué formato (de entre al menos 2 posibles) quiere los informes resultantes del análisis ✓

Obtención de los informes en formatos distintos.

El informe que proporciona Analog, como resultado de cada análisis que realiza, es bastante intuitivo y fácil de comprender. Se posibilita en dos formatos diferentes: texto o HTML, siendo este último el más utilizado por la mayoría de los usuarios, ya que permite visualizar los resultados con cualquier navegador web y configurarlo mediante hojas de estilo y gráficos personalizados.

Además, Analog puede exportar los informes en varios formatos distintos incluidos HTML, XHTML, XML, Latex y en un modo de salida determinado (como los CSV) dándole al usuario final la oportunidad de poder importarlos a otros programas que les permitan, por ejemplo, generar informes más estructurados y enriquecidos en gráficos.

- d) La aplicación ha de permitir el envío de los resultados del análisis por e-mail ✗

Envío de los informes por correo electrónico

Gracias a los dos tipos de formato de salida que proporciona, si bien no es una opción directa de la aplicación el envío directo de los informes por e-mail, sí que puede el usuario enviar posterior y personalmente los resultados obtenidos, comprimidos, vía email.

- e) Herramienta configurable por el usuario: El usuario ha de poder configurar los informes proporcionados: tipos de informe a obtener ✓, parámetros para el análisis ✗ y datos que componen los informes resultantes ✓

¿Qué puede hacer el usuario con los datos de entrada?

Analog es altamente configurable, permitiendo al usuario elegir entre 200 opciones distintas de configuración posibles, lo que hace que la herramienta parezca muy complicada de usar para un usuario de a pie, normal y corriente. Pero, si nos ponemos en el papel de un experto en el dominio, este “problema” desaparece y pasar a ser una gran ventaja para ellos, ya que transforma Analog en una herramienta muy útil y versátil.

El hecho de que Analog no presente una GUI de usuario, y que la interacción con la herramienta sea a través de la línea de comandos, puede resultar poco amigable para el usuario, por lo que para su configuración Analog proporciona además una página HTML desde donde se pueden marcar las características principales de la aplicación directamente (en español, la página es `esform.html`):

Interfaz de Formularios de Analog

1. Selección de Informes

Consulte [la página principal de analog](#) para la descripción de los distintos informes.

Que informes desea ver ?

<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Resumen General
<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Informe Mensual
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe Semanal
<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Resumen Diario
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe Diario
<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Resumen por Horas
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Dominios
<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Informe por Organizaciones
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Directorios
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por tipo de Archivos.
<input checked="" type="radio"/> [Si]	<input type="radio"/> [No]	Informe por Peticiones
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Tamaño de Archivos
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Remitentes
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Consultas de Búsquedas
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Búsquedas de palabras.
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Resumen por Navegador.
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Sistema Operativo.
<input type="radio"/> [Si]	<input checked="" type="radio"/> [No]	Informe por Código de Retorno

Ahora ya puede ejecutar el programa: [Producir Estadísticas](#)

O puede completar las siguientes opciones para informes individuales. Puede usar bytes para indicar 10 Megabytes; también para indicar los 50 ítems con más bytes.

2. Opciones para el informe detallado

Opciones del Informe por Dominios

Ordenar el Informe por Dominios

Incluir todos los dominios con al menos bytes

Opciones del Informe por Organizaciones

Ordenar el Informe por Organizaciones

Incluir todas las Organizaciones con al menos peticiones en total

Opciones del Informe por Directorios

Ordenar el Informe por Directorios

Incluir todos los directorios con al menos bytes

Opciones del Informe por Peticiones

Ordenar el Informe por Peticiones

Incluir todos los archivos con al menos peticiones

Mostrar

Opciones del Informe por Remitentes
 Ordenar el Informe por Remitentes
 Incluir todos los Remitentes con al menos

3. Analizando sólo una parte del archivo de registro

Sólo ciertas fechas
 Puede analizar sólo las peticiones entre ciertas fechas. Ingrese el rango de fechas más abajo de la siguiente manera aammdd; ej., 980301 para el 1ro de Marzo de 1998 (o llene sólo una casilla para limitar el rango de fechas en un solo sentido).
 Desde hasta

Sólo ciertos archivos
 Sólo considerar los siguientes archivos (lista, separada por comas; puede contener el caracter comodín *)

 Ignorar los siguientes archivos

4. Presentación

 El nombre de su organización (para el título de la página)

 La página principal de su Organización
 URL:

Ilustración 26: Analog - Página de configuración de Informes

Es justamente en esta página donde se recogen la mayoría de las opciones que va a tener en cuenta el programa a la hora de realizar los informes – opciones como la periodicidad del análisis que se requiere o los intervalos de análisis a tener en cuenta.

Analog es el tipo de herramienta que no usan dos usuarios de la misma manera, debido especialmente al gran número de opciones de configuración que presenta. Por lo que, muchas de sus funcionalidades suelen perderse durante el camino – siendo muy difícil, por una sola persona, llegar a conocer y usar todas ellas con soltura.

Gracias al fichero de configuración de la herramienta, podemos obtener diferentes combinaciones de análisis de datos, pero no podemos un peso/valor a los distintos parámetros analizados, por lo que el usuario no tiene la opción de modificar la lógica de análisis del sistema según sus necesidades, sino, únicamente la configuración de los informes a realizar.

- f) **La aplicación debe contar con una interfaz de usuario que facilite la interacción con el sistema** ✖

Interfaz de Usuario

Si bien es cierto que no presenta una interfaz de usuario propiamente dicha, gracias a los formularios para su configuración, no es que haga especialmente falta, ya que éstos, unidos al formato de informe que proporciona la herramienta, su claridad de exposición e interpretación y su perfecta navegabilidad a través de los mismos, podrían llegar a cubrir, más o menos, las funcionalidades básicas deseadas

2.3 Tabla Comparativa

A continuación se expone una tabla comparativa con las distintas soluciones similares presentadas, donde:

- **C1** es la Característica Buscada 1 = Analizar un fichero de *logs*.
- **C2** es la Característica Buscada 2 = Mostrar resultados por pantalla.
- **C3** es la Característica Buscada 3 = Poder obtener los informes en formatos distintos.
- **C4** es la Característica Buscada 4 = Poder enviar los resultados por correo electrónico.
- **C5** es la Característica Buscada 5 = Tipos de Informes Configurables por el usuario.
- **C6** es la Característica Buscada 6 = Parámetros de análisis Configurables por el usuario.
- **C7** es la Característica Buscada 7 = Datos de los Informes Configurables por el usuario.
- **C8** es la Característica Buscada 8 = Interfaz de Usuario.

	C1	C2	C3	C4	C5	C6	C7	C8
Event Log Explorer	✓	✓	✓	✗	✓	✗	✓	✓
Performance Analysis of Logs	✓	✓	✓	✗	✓	✓	✓	✓
Log Parser Studio	✓	✓	✓	✗	✓	✗	✓	✗
Analog	✓	✓	✓	✗	✓	✗	✓	✗

Tabla 4: Tabla Comparativa de las Soluciones Similares Propuestas

Si bien es cierto que todas estas herramientas cumplen gran parte de las características buscadas, ninguna de ellas llega a cumplirlas todas. Y, si además tenemos en cuenta que no hemos encontrado ningún sistema que sea capaz de analizar el tipo de fichero requerido - un fichero de *logs* generado con la herramienta Snoopy [1], con una configuración específica de formato de registros, que realice el análisis requerido bajo las especificaciones marcadas, queda totalmente argumentada la necesidad de realizar el proyecto de la presente propuesta.

Estudio de Viabilidad, Análisis, Diseño e Implementación

Metodología usada, ciclo de vida, marco regulador, etc.

Para el desarrollo de estos apartados, que constituyen el cuerpo principal de la memoria, se ha decidido seguir una adaptación de Métrica v.3 [12], una metodología de desarrollo del software muy completa y conocida que ayuda a las organizaciones actuales a sistematizar las actividades que dan soporte al ciclo de vida del software.

De esta manera, los tres siguientes puntos los haremos siguiendo el proceso de *Desarrollo de Sistemas de Información* presentado en la estructura principal de Métrica v.3 [12] que contiene todas las actividades y tareas que deben llevarse a cabo para realizar un buen desarrollo del sistema, cubriendo: desde el análisis de requisitos hasta la instalación del software (únicamente se tendrán en cuenta aquellas actividades y tareas que apliquen al caso de estudio actual).

3. Estudio de Viabilidad del Sistema

DESCRIPCIÓN Y OBJETIVOS

De acuerdo a Métrica v.3 [12] el Estudio de Viabilidad del Sistema tiene como objetivo el análisis del conjunto concreto de necesidades para el que se desea proponer una solución a corto plazo, a fin de tener en cuenta las restricciones económicas, técnicas, legales y operativas que condicionarán nuestro trabajo.

Las actividades que hemos realizado en este estudio son las mismas que se contemplan en la guía de Métrica v.3 [12]:

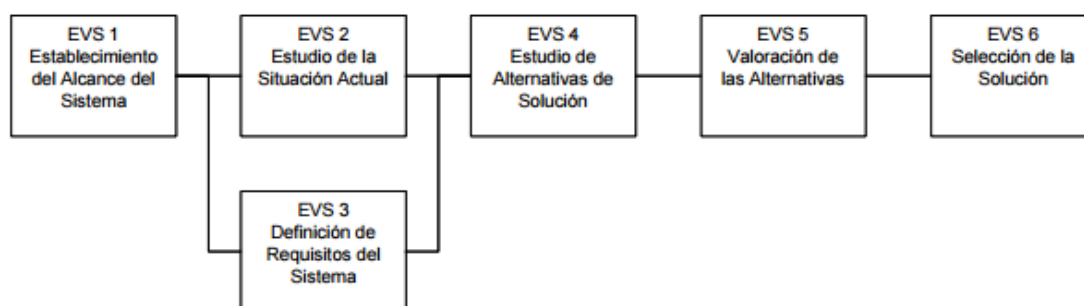


Ilustración 27: Actividades del Estudio de Viabilidad del Sistema de Métrica v.3
Fuente: Métrica v.3 [12] - Gobierno de España

Como se muestra en esta ilustración, tomada de Métrica v.3 [12], el estudio termina con la selección de una solución (EVS 6) entendida como el resultado del estudio (EVS4) y valoración (EVS 5) de alternativas que, satisfaciendo los requisitos del sistema (EVS 3), permiten pasar de una situación actual (EVS 2) a una situación deseada (EVS 1) .

En este capítulo, partiendo del estado inicial, la situación actual y los requisitos planteados, estudiamos las alternativas de solución que incluyen tanto desarrollos a medida como uso de productos software del mercado.

Una vez descritas cada una de las alternativas planteadas, se valora la inversión a realizar en cada caso, los riesgos asociados y su impacto en la organización con el fin de que el cliente pueda evaluar las distintas alternativas y seleccionar la más adecuada.

3.1 Establecimiento del Alcance del Sistema

Este apartado se corresponde con la Actividad EVS 1 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

Como primera actividad hemos estudiado el alcance de la necesidad que afronta este proyecto realizando las 3 tareas descritas en Métrica v.3 [12].

En la tarea EVS 1.1 Estudio de la Solicitud se estudia la necesidad planteada por el proyecto, definiéndola y teniendo en cuenta sus condicionantes principales, para concluir con los Objetivos generales del sistema a desarrollar.

En la tarea EVS 1.2 Identificación del Alcance del Sistema se estudia el alcance de la necesidad planteada por el proyecto, situándola, en este caso, al margen de otros posibles proyectos existentes dentro de la Planificación de Sistemas de Información de la entidad.

En la tarea EVS 1.3 Especificación del Alcance del EVS se identifican las unidades organizativas afectadas estableciendo su estructura y los perfiles de los usuarios (Catálogo de Usuarios).

3.1.1 Estudio de la Solicitud

(Métrica v.3 [12] Tarea EVS 1.1)

Definición

La necesidad que da lugar a este proyecto es contar con una herramienta que permita realizar el análisis de un fichero *log* que recoge los comandos UNIX utilizados por los usuarios en un entorno de trabajo compartido, en concreto las aulas informáticas de la universidad, y que sea aplicable para situaciones similares.

Es decir, que el usuario pueda seleccionar el fichero *log* concreto que quiere analizar en el sistema y que pueda seleccionar dentro de él los aspectos que considere importantes y presentarlos (informes) en el formato deseado.

Condicionantes principales

Dado que los ficheros de *logs* suelen contener gran cantidad de registros, uno de los condicionantes que hemos tenido en cuenta en la aplicación es el tamaño de los ficheros con los que se va a trabajar por el tiempo de procesamiento que esto implica y que es necesario tener presente a la hora del diseño.

Otro de los condicionantes que hemos encontrado relevante es la restricción legal impuesta por la Ley de Protección de datos de carácter personal, conforme a la Ley Orgánica 15/1999 [13].

El resto de condicionantes de tipo económico, tecnológico y operativo, son los típicos de un proyecto informático normal por lo que no hemos encontrado en ellos ningún tipo de restricción que nos haga desistir de la propuesta.

En consecuencia, concluimos esta tarea con una Descripción general de la necesidad planteada en el estudio y sus restricciones, que presentamos como:

Objetivos Generales del Sistema:

- Analizar un fichero de *logs* que contenga los comandos utilizados por los usuarios de un sistema Unix.
- Permitir al usuario seleccionar el fichero de *logs* que desea analizar.
- Facilitar el reinicio de la base de datos y el borrado de todos los datos almacenados en ella, fruto de análisis anteriores.
- Proporcionar al usuario la posibilidad de seleccionar el tipo de informe que desea obtener.
- Reportar los informes resultantes del análisis realizado en un documento en formato Excel o PDF, según lo especifique el usuario.
- Posibilitar el envío de los reportes obtenidos a una dirección de correo electrónico proporcionada por el usuario, si éste así lo cree conveniente.
- Reiniciar la base de datos y recargar la información inicial con los parámetros que el usuario considere de su interés, es decir, permitir al usuario cambiar los pesos asignados a las variables de estudio para cada uno de los comandos que quiere analizar.
- Gestionar los datos y las operaciones en forma tal que los tiempos de procesamiento sean los adecuados a juicio de los usuarios

- Respetar la legislación vigente en cuanto a la protección de datos de carácter personal
- Analizar un fichero de *logs* que contenga los comandos utilizados por los usuarios de un sistema Unix.
- Permitir al usuario seleccionar el fichero de *logs* que desea analizar.
- Facilitar el reinicio de la base de datos y el borrado de todos los datos almacenados en ella, fruto de análisis anteriores.
- Proporcionar al usuario la posibilidad de seleccionar el tipo de informe que desea obtener.
- Reportar los informes resultantes del análisis realizado en un documento en formato Excel o PDF, según lo especifique el usuario.
- Posibilitar el envío de los reportes obtenidos a una dirección de correo electrónico proporcionada por el usuario, si éste así lo cree conveniente.
- Reiniciar la base de datos y recargar la información inicial con los parámetros que el usuario considere de su interés, es decir, permitir al usuario cambiar los pesos asignados a las variables de estudio para cada uno de los comandos que quiere analizar.

3.1.2 Identificación del Alcance del Sistema

(Métrica v.3 [12] Tarea EVS 1.2)

El sistema a desarrollar no es consecuencia de la realización de un PSI (Plan del Sistema de Información) de la Universidad, ni de ninguna entidad; ni constituye la sustitución de ninguno anterior existente, ni supone ningún complemento adicional a alguno que ya se está utilizando actualmente, sino que establece una herramienta totalmente. No presenta por tanto restricciones particulares relativas a la sincronización con otros proyectos que puedan interferir en la planificación y futura puesta a punto del sistema objeto del estudio.

Su alcance viene limitado solo por las especificaciones indicadas por el cliente y las que puedan derivarse de la situación actual.

3.1.3 Especificación del Alcance del EVS

(Métrica v.3 [12] Tarea EVS 1.3)

Dado que el alcance del sistema a desarrollar se extiende únicamente a las unidades organizativas afectadas por el sistema, el alcance específico del EVS se extiende a los siguientes roles o perfiles a quienes afecta directamente y quienes pueden influir en el éxito o fracaso del mismo.

- Cliente: Persona interesada en la puesta en marcha del proyecto. Es quien ha concebido la idea de nuestro sistema y su necesidad de implementación.
- Ingeniero y/o equipo de desarrollo: Persona, o grupo de personas, responsables del desarrollo completo del sistema. Su objetivo principal será generar un producto software que cubra las necesidades del cliente.
- Usuario/s: Persona o grupo de personas que utilizarán el sistema una vez esté totalmente implementado. En nuestro caso, creemos conveniente distinguir entre
 - usuario Administrador (capaz de gestionar y actualizar las funcionalidades básicas del sistema),
 - y usuario Normal (que será aquel que utilizará la herramienta única y exclusivamente para su fin, es decir, que la utilizará tal y como se presenta, sin hacerle ningún tipo de modificación o cambio que la pudiera adaptar mejor a sus necesidades).

3.2 Estudio de la Situación Actual

Este apartado se corresponde con la Actividad EVS 2 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

Como segunda actividad hemos realizado un diagnóstico global de la situación actual por cuanto, tal como lo indica Métrica v.3 [12] hemos decidido no describir la situación actual por los siguientes motivos:

La situación actual es el estado en el que se encuentran los sistemas de información existentes en el momento en el que se inicia su estudio.

En nuestro caso no hemos encontrado en la Universidad ningún sistema de información existente que cumpla con los objetivos establecidos en el presente estudio ni ningún Plan de Sistemas de Información que contemple su implementación o la descarte.

No hay, por tanto, ningún sistema de información concreto existente que se encuentre implicado en este proyecto y que, por consiguiente, sea necesario analizar

previamente, es decir, realizar sobre él las 4 tareas descritas en v.3: valorarlo (EVS 2.1), identificar sus usuarios (EVS 2.2), describirles y localizarles (EVS 2.3) e identificar sus problemas o posibles mejoras (EVS 2.4).

Diagnóstico global

La situación actual que nos encontramos es que efectivamente hay un uso importante de comandos por parte de los usuarios de sistemas Unix, que no controlamos.

Si bien es cierto que existen aplicaciones y comandos para almacenarlos y mostrarlos al usuario sin tratamiento alguno, no hemos encontrado ninguna herramienta que proporcione al usuario la posibilidad de reportarle un análisis completo de qué comandos se están utilizando más, quién los está utilizando, si se está utilizando un comando determinado, cuántas veces, etc. Esto lo tendría que hacer el propio administrador del sistema, o en su defecto, el usuario interesado, totalmente a mano, de forma arcaica, yendo línea a línea en los miles, o millones de registros que pueda haber en el fichero o ficheros de *log* que contengan la información que está buscando.

Por ello, podemos afirmar que se trata de un sistema de información totalmente novedoso en lo que a análisis de *logs* de comandos Unix se refiere; y partiremos directamente de cero haciendo un análisis exhaustivo de las necesidades del cliente y plasmándolas en un catálogo de requisitos que nos ayude desarrollar el producto que queremos llevar a cabo. Haremos un análisis y diseño que se adapte, en la medida de lo posible, a las especificaciones requeridas. Y nos serviremos, si se puede, de productos existentes en el mercado que nos ayuden a cumplir alguna de las necesidades del usuario (total o parcialmente).

3.3 Definición de los Requisitos del Sistema

Este apartado se corresponde con la Actividad EVS 3 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

En esta actividad hemos determinado los requisitos generales y definido sus prioridades, en base a una serie de sesiones de trabajo planificadas y realizadas con el cliente y los usuarios participantes. Como resultado hemos configurado un catálogo de requisitos con el que analizar y valorar las distintas alternativas de solución.

Dado que el sistema en estudio no pertenece al ámbito de un Plan de Sistemas de Información no ha sido necesario realizar la tarea EVS 3.1 Identificación de las Directrices Técnicas y de Gestión por cuanto no hay un marco de referencia a considerar en esta tarea; pasando directamente a las tareas EVS 3.2 Identificación de Requisitos y EVS 3.3 Catalogación de Requisitos sobre la base de lo realizado en la tarea EVS 1.2.

En la tarea EVS 3.2 Identificación de Requisitos se ha identificado las necesidades que ha de cubrir el sistema de acuerdo a lo transmitido por el cliente.

En la tarea EVS 3.3 Catalogación de Requisitos se ha formalizado la información sobre cada uno de esos requisitos y constituido con ellos un catálogo que permita comprobar que las alternativas a considerar cumplen con la necesidad del cliente.

3.3.1 Identificación de los Requisitos

(Métrica v.3 [12] Tarea EVS 3.2)

En esta tarea, se persigue crear un catálogo de requisitos que cubra todas las necesidades que ha de cubrir el sistema en estudio, de acuerdo a lo transmitido por el cliente en las distintas reuniones llevadas a cabo.

Para facilitar la identificación de los requisitos hemos distinguido entre requisitos funcionales y no funcionales.

Como requisitos funcionales (requisitos de capacidad) entendemos aquellas especificaciones que indican funcionalidades que el usuario podrá realizar con el sistema; es decir, lo que el usuario quiere poder hacer con la aplicación.

Dentro de esta categoría hemos identificado los siguientes:

1. RU-C01 Borrar la información almacenada en la base de datos.
2. RU-C02 Cargar los comandos iniciales.
3. RU- C03 Importar el ficho de *logs*.
4. RU-C04 Analizar el fichero de *logs*.
5. RU-C05 Visualizar los resultados del Análisis.
6. RU-C06 Realizar informes con los resultados.

Como requisitos no funcionales (requisitos de restricción) entendemos aquellas especificaciones que indican los límites o restricciones a implementar en el sistema; es decir, las restricciones impuestas por los usuarios sobre cómo se debe resolver un problema o cómo se deben alcanzar los objetivos recogidos en los requisitos de capacidad.

Dentro de esta categoría hemos identificado los siguientes:

1. RU-R01 El acceso será libre (no habrá control de acceso).
2. RU-R02 Se podrá analizar todo fichero de *log* capturado con la herramienta Snoopy.
3. RU-R03 Los informes se mostrarán en formato PDF con opción de una copia en formato PDF y/o Excel.
4. RU-R04 Se podrán enviar los informes obtenidos por e-mail.
5. RU-R05 Dispondrá de una interfaz gráfica que facilite la interacción al usuario.
6. RU-R06 Permitirá usar como lenguaje de interacción el Español y/o el Inglés.

La información obtenida en las sesiones de trabajo se ha definido de forma tal que permite analizar y valorar en base a ella las distintas alternativas de solución.

La información sobre cada requisito se ha descrito en una tabla con los siguientes campos informativos:

- **Identificador:** código unívoco para identificar cada uno de los requisitos expuestos. Seguirá la regla de nombrado RU-TXX, donde T tomará los valores C (para los requisitos de usuario de capacidad) o R (para los requisitos de usuario de restricción) y X e Y conformarán un valor numérico de dos cifras que funcionará como un contador incremental para cada uno de los grupos de requisitos de usuario existentes, empezando siempre por el 01.
- **Nombre:** Título descriptivo del requisito que nos indicará de forma breve y concisa en qué consiste y de qué trata.
- **Claridad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica cuán claro es el requisito en sí mismo, es decir, si tiene una única interpretación, sin ningún tipo de ambigüedad.
- **Fuente:** Nos indica si el requisito ha sido originado gracias a la interacción con el cliente o si proviene de otros medios.
- **Necesidad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica la necesidad que tiene de implementarse dentro del producto final.
- **Prioridad:** Puede tomar los valores: “baja”, “media” o “alta” y nos indica la prioridad que tiene el requisito dentro del sistema, es decir, el orden que deberían ocupar en el desarrollo de los distintos módulos del sistema.
- **Verificabilidad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica si es posible verificar de forma concreta si el requisito ha sido incorporado en el producto final.
- **Estabilidad:** Podrá tomar los valores “Estable” e “Inestable” y nos indica si un requisito es susceptible o no a cambios a lo largo del desarrollo del proyecto.
- **Descripción:** Este campo es totalmente variable, y su único objetivo es el de describir detalladamente el requisito en sí.

Identificador: RU-TXY				
Nombre	Nombre del Requisito.			
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input type="checkbox"/> Otros	
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional			
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Estabilidad	Estabilidad del Requisito durante el desarrollo		<input type="checkbox"/> Estable <input type="checkbox"/> Inestable	
Descripción	Descripción breve de las especificaciones del requisito.			

Tabla 5: Especificación del formato que seguirá nuestro catálogo de requisitos de usuario

3.3.2 Catalogación de Requisitos

(Métrica v.3 [12] Tarea EVS 3.2)

Para catalogar los requisitos de usuario que vamos a plasmar en este documento, utilizaremos básicamente dos grupos generales de los mismos distinguiendo entre requisitos de capacidad, o requisitos funcionales, y requisitos de restricción, o no funcionales.

3.3.2.1 Requisitos de Usuario de Capacidad

Se entenderán como requisitos de capacidad, aquellas especificaciones que indiquen las funcionalidades que el usuario podrá realizar con el sistema, o dicho de una forma más sencilla, lo que el usuario quiere poder hacer con la aplicación.

Identificador: RU-C01			
Nombre	Borrar Base de Datos		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema permitirá el borrado de la información almacenada en la base de datos.		

Tabla 6: Requisito de usuario de capacidad 01 (RU-C01)

Identificador: RU-C02			
Nombre	Cargar Comandos Iniciales		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema permitirá el cambio de la lista de comandos iniciales, base de la lógica para realizar los análisis.		

Tabla 7: Requisito de usuario de capacidad 02 (RU-C02)

Identificador: RU-C03			
Nombre	Cargar Fichero de <i>Logs</i>		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial ■ Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta ■ Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	El sistema permitirá la importación del fichero <i>log</i> que se quiere analizar por parte del usuario.		

Tabla 8: Requisito de usuario de capacidad 03 (RU-C03)

Identificador: RU-C04			
Nombre	Analizar fichero de <i>logs</i>		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	El sistema realizará el análisis de la información contenida en un fichero de <i>logs</i> determinado.		

Tabla 9: Requisito de usuario de capacidad 04 (RU-C04)

Identificador: RU-C05			
Nombre	Visualizar Resultados del Análisis		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	El sistema ha de permitir al usuario visualizar los resultados del análisis realizado.		

Tabla 10: Requisito de usuario de capacidad 05 (RU-C05)

Identificador: RU-C06			
Nombre	Tipos de Informe		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		
	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema debe presentar al usuario la posibilidad de realizar al menos dos tipos de informes diferentes.		

Tabla 11: Requisito de usuario de capacidad 06 (RU-C06)

3.3.2.2 Requisitos de Usuarios de Restricción

Se entenderán como requisitos de restricción, aquellas especificaciones que indiquen los límites o restricciones a implementar en el sistema, es decir, las restricciones impuestas por los usuarios sobre cómo se debe resolver un problema o cómo se deben alcanzar los objetivos recogidos en los requisitos de capacidad.

Identificador: RU-R01			
Nombre	Usuarios de la Aplicación		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		
	<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	La aplicación será de libre acceso, es decir, no habrá ningún mecanismo de control de acceso y se uso se realizará siempre bajo el perfil de un único tipo de usuario: administrador.		

Tabla 12: Requisito de usuario de restricción 01 (RU-R01)

Identificador: RU-R02			
Nombre	Formato del Fichero de Logs		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema ha de ser capaz de analizar un fichero de <i>log</i> capturado con la herramienta Snoopy [1].		

Tabla 13: Requisito de usuario de restricción 02 (RU-R02)

Identificador: RU-R03			
Nombre	Entrega de los informes Realizados		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema mostrará al usuario los informes obtenidos en formato PDF y además, le proporcionará la opción de poder conseguir una copia de los mismos en formato PDF y/o Excel.		

Tabla 14: Requisito de usuario de restricción 03 (RU-R03)

Identificador: RU-R04			
Nombre	Enviar Informes por Correo Electrónico		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input checked="" type="checkbox"/> Inestable		
Descripción	El sistema podrá permitir al usuario el envío de los informes obtenidos por e-mail.		

Tabla 15: Requisito de usuario de restricción 04 (RU-R04)

Identificador: RU-R05			
Nombre	Interfaz Gráfica		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial ■ Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta ■ Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	La aplicación podrá tener una interfaz gráfica que facilite la interacción al usuario.		

Tabla 16: Requisito de usuario de restricción 05 (RU-R05)

Identificador: RU-R06			
Nombre	Idioma de la interfaz		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial ■ Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media ■ Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input type="checkbox"/> Estable ■ Inestable
Descripción	La interfaz del sistema podrá proporcionar al usuario al menos dos lenguajes distintos de interacción: Español / Inglés		

Tabla 17: Requisito de usuario de restricción 06 (RU-R06)

3.4 Estudio de Alternativas de Solución

Este apartado se corresponde con la Actividad EVS 4 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

Bajo esta actividad hemos estudiado diversas alternativas de solución acordes con los requisitos planteados, realizando las 2 tareas descritas en Métrica v.3 [12].

En la tarea EVS 4.1 Preselección de Alternativas de Diseño se estudian las diferentes opciones que hay para configurar la solución. Entre ellas, hemos considerado tanto la adquisición de productos estándar del mercado, como desarrollos a medida o soluciones mixtas.

En la tarea EVS 4.2 Descripción de las Alternativas de Diseño se establecen como alternativas de diseño a evaluar cuatro combinaciones de las opciones elegidas para satisfacer los requisitos del cliente y en función de alguna característica que puede resultar relevante para él.

3.4.1 Preselección de Alternativas de Solución

(Métrica v.3 [12] Tarea EVS 3.2)

Dados los requisitos a cubrir por el sistema, en esta tarea estudiamos las diferentes opciones que tenemos para configurar la solución, que sirvan tanto para la adquisición de productos software estándar del mercado como para desarrollos a medida o soluciones mixtas.

Entre estas opciones consideramos:

- como alternativas de sistema operativo
 - o Windows
 - o Linux
- como alternativas de diseño
 - o diseño modular (dividiendo el sistema en subsistemas)
 - o diseño monolítico (manteniendo el sistema como un único todo)
- como alternativas de lenguaje de programación
 - o Java [14] en entorno Eclipse [15] o Netbeans [16]
 - o C# [17] en entorno Visual Studio.NET [18]
 - o Java EE [19] con un servidor Apache Tomcat [20]
- como alternativas para el gestor de base de datos:
 - o Microsoft Access [22]
 - o Oracle [23]
 - o MySQL [25]
 - o PostgreSQL [27]

De cada una de estas opciones analizamos sus características y/o ventajas e inconvenientes en los siguientes apartados.

3.4.1.1 Alternativas de diseño

Diseño Modular

Ventajas:

En el diseño modular se divide el sistema en subsistemas completos con funcionalidades específicas separadas entre sí.

- Los cambios y/o evoluciones dentro de cada subsistema afectarán únicamente a este subsistema, haciendo más fácil su detección, localización y subsanación.
- Las funcionalidades específicas de cada módulo pueden ser reutilizadas en forma separada en otro entorno sin necesidad de reutilizar todo el sistema, sino solo el módulo correspondiente.

Desventajas:

La división del sistema en módulos implica normalmente mayor tiempo de desarrollo y ejecución.

- de desarrollo porque cada uno de los módulos deberá ser completo por sí mismo de modo que pueda ser reutilizado en cualquier otro entorno
- de ejecución porque los módulos deben interactuar entre sí para llevar a cabo la funcionalidad general del sistema, lo que añade tiempos de comunicación entre ellas que afectan al tiempo total necesario para su ejecución

Diseño Monolítico

Ventajas:

En el diseño monolítico se mantiene el sistema como un todo único lo que hace que la funcionalidad general del sistema dependa en exclusiva de él; es decir, de él depende enteramente el leer los datos, procesarlos, analizarlos y mostrar los resultados al usuario final.

Esto elimina la necesidad de los tiempos adicionales considerados en la alternativa modular, lo que hace que los tiempos de ejecución y desarrollo se reduzcan.

Desventajas:

Al estar toda la funcionalidad junta en un único módulo,

- cualquier cambio o evolución que se produzca en él afectará al funcionamiento general siendo más difícil su detección, localización y subsanación;
- e impide la reutilización separada de funcionalidades específicas, sólo podríamos llevarnos la lógica para volver a implementarla nuevamente de cero en el nuevo sistema.

Conclusión:

La forma monolítica de diseño es adecuada para proyectos pequeños que no vayan a sufrir grandes cambios o evoluciones a lo largo de su vida. La forma modular es más adecuada para el mantenimiento y desarrollo de la herramienta durante toda su etapa evolutiva, siendo la aconsejable en proyectos grandes. En nuestro, el sistema a desarrollar ocupa una posición intermedia que le permite sacar ventaja de cualquiera de las dos opciones, por lo que consideramos ambas como posibles.

3.4.1.2 Alternativas de Lenguaje de Programación

Para la codificación de todas las funcionalidades requeridas por nuestro sistema: desarrollo hecho a medida, interfaz de usuario, etc. hemos tenido en cuenta tres alternativas de lenguaje de programación.

1) Java [14] en entorno Eclipse [15] o Netbeans [16]

Java [14] es un lenguaje de programación orientado a objetos muy robusto, portable, totalmente gratuito e independiente de la plataforma de desarrollo, que puede funcionar indistintamente en cualquier entorno y en cualquier dispositivo hardware donde pueda ejecutarse su JRE ^[1].

Eclipse [15] es un entorno de desarrollo libre y gratuito, personalizable a través de *plug-ins* que le dan mayor funcionalidad. Es totalmente portable.

Netbeans [16] es un entorno de desarrollo libre y gratuito, sin restricciones de uso, muy parecido a Eclipse [15], pero un poco más intuitivo y fácil de manejar, sobre todo en lo que a la realización de interfaces gráficas se refiere.

^[1] Java Runtime Environment o JRE es la parte del kit de desarrollo de java que constituye el conjunto de utilidades necesarias que permite la ejecución de programas Java en distintos dispositivos hardware.

2) C# [17] en entorno Visual Studio .NET [18]

C# [17] es el lenguaje de programación orientado a objetos de Microsoft. Está diseñado para permitir a los desarrolladores compilar sus aplicaciones en diferentes dispositivos hardware gracias a su .Net Framework. Actualmente es un lenguaje multiplataforma, aunque está pensado principalmente para funcionar en conjunto con Windows, por lo que es mejor realizarlo bajo este Sistema Operativo.

Visual Studio [18] es el entorno de desarrollo de Microsoft para lenguajes .Net. Tiene una versión gratuita de 90 días y/o para estudiantes, pero si se quiere usar para entornos profesionales, hay que pagar una licencia que puede ir desde 387€ hasta 3328€, dependiendo de las necesidades del cliente.

3) Java EE [19] con un servidor Apache Tomcat [20]

Java EE [19] es una plataforma de programación gratuita hecha principalmente para llevar a cabo desarrollos de aplicaciones portables, escalables y seguras del lado del servidor. Es prácticamente un estándar para implementar aplicaciones web y aplicaciones con arquitectura SOA (orientada a servicios). Proporciona una arquitectura multicapa, destacando principalmente las capas: Cliente / Servidor o Negocio / Datos.

Esta opción se ha considerado para tener una opción de desarrollo web.

Tomcat [20] sería el contenedor de *servlets* que utilizaríamos para nuestro sistema, ya que implementa las especificaciones de los *servlets* y JSP's ^[1] de Java [14].

En el caso de decidir utilizar la opción web, la interfaz gráfica se haría en PHP [21], pues es un lenguaje de programación del lado del servidor especialmente diseñado para ello.

3.4.1.3 Alternativas para la Base de Datos

Las herramientas de gestión y/o almacenamiento de datos que hemos considerado para conseguir las funcionalidades que necesitamos para nuestro sistema han sido las siguientes:

1) Microsoft Access [22]

Es el sistema de Gestión de Bases de Datos Relacionales de Microsoft. Está incluido en el paquete básico del Office y ha sido creado principalmente para uso personal y de pequeñas organizaciones. Es una herramienta de pago.

^[1] JSP (Java Server Pages) Tecnología para la creación de páginas web dinámicas basadas en XML y HTML entre otros. 79

2) Oracle [23]

Es un sistema de Gestión de Base de Datos Objeto-Relacional desarrollado por Oracle Corporation [23]. Las características principales por las que se le considera uno de los sistemas de bases de datos más completos es su estabilidad, escalabilidad y soporte multiplataforma y de transacciones. Es también una herramienta de pago.

3) MySQL [[25]

Es un sistema de Gestión de Bases de Datos relacional, multihilo y multiusuario. Tiene licenciamiento dual, es decir, tiene una versión libre bajo la licencia GNU GPL ^[1] para uso puramente interno, pero si se le quiere dar un uso más amplio, habría que pagar una licencia específica para tal fin.

4) PostgreSQL [27].

Es una base de datos objeto-relacional (igual que Oracle [23]) y multiprocesos en vez de multihilos (como MySQL [25]) totalmente gratuita.

3.4.2 Descripción de las Alternativas de Solución

(Métrica v.3 [12] Tarea EVS 4.2)

En esta tarea hemos combinado las alternativas disponibles para el gestor de base de datos, diseño y lenguaje de programación para el desarrollo del sistema y su interfaz de cuatro formas distintas a fin de configurar otras tantas alternativas de solución completas para su evaluación final.

Todas las combinaciones cumplen con los requisitos solicitados por el cliente para el sistema y han sido conformadas pensando en una característica específica que las distingue claramente unas de otras:

La Alternativa 1 está constituida por herramientas y aplicaciones gratuitas o de código libre.

La Alternativa 2 está constituida por productos populares del mercado desarrollados por empresas comerciales diversas.

La Alternativa 3 está constituida por herramientas y aplicaciones de Microsoft.

La Alternativa 4 está constituida por herramientas y aplicaciones del mercado que permiten implementar el sistema como una aplicación web.

Alternativas		Sistema Operativo	Tipo de Diseño	Lenguaje de Programación	Gestor de Base de datos
	1	Linux	Monolítico	<u>Java</u> con entorno de desarrollo <u>Eclipse</u>	MySQL
	2	Windows	Modular	<u>C#</u> en entorno <u>Visual Studio.Net</u>	Oracle
	3	Windows	Monolítico	<u>Java</u> con entorno de desarrollo <u>Netbeans</u>	Microsoft Access
	4	Windows	Modular	<u>Java EE</u> con un servidor Apache <u>Tomcat</u> .	Oracle

Tabla 18: Comparativa Alternativas de Solución Propuestas

ALTERNATIVA 1:

- Sistema Operativo Linux.
- Tipo de diseño monolítico.
- Lenguaje de programación: Java con entorno de desarrollo Eclipse.
- Gestor de Base de datos: MySQL

Se trata de una alternativa básica, pero suficiente, que se ha confeccionado buscando herramientas y aplicaciones gratuitas o de código libre, que permitan conseguir los requisitos de nuestro sistema minimizando los costes de las licencias.

Se incluyó MySQL [25] y no PostgreSQL [27] porque en principio el cliente quiere la aplicación para uso interno, aunque con posibilidad futura de una ampliación de uso mayor. Si no hubiese sido por dejar abierta esta posibilidad la opción lógica hubiese sido PostgreSQL [27].

ALTERNATIVA 2:

- Sistema Operativo Windows.
- Tipo de diseño modular.
- Lenguaje de programación: C# en entorno Visual Studio .Net.
- Gestor de Base de Datos: Oracle.

Se trata de una alternativa básica, pero suficiente, que se ha confeccionado buscando herramientas y aplicaciones populares en el mercado, lo que implica una reducción en el periodo de adaptación y aprendizaje por parte del cliente.

ALTERNATIVA 3

- Sistema Operativo Windows.
- Tipo de diseño monolítico.
- Lenguaje de programación: Java con entorno de desarrollo Netbeans.
- Gestor de Base de Datos: Microsoft Access.

Se trata de una alternativa similar a la anterior con la diferencia que en este caso se usan únicamente herramientas y aplicaciones de Microsoft.

Como ya quedó reseñado en el apartado anterior, Microsoft Access [22] es un sistema gestor apropiado para uso personal y pequeñas organizaciones; mientras que Oracle [23] o MySQL [25] se adaptan mejor a las grandes organizaciones por la velocidad de procesamiento, la seguridad de los datos y el uso de procedimientos almacenados y *triggers*.

El carácter intermedio del uso que el cliente quiere dar al sistema a desarrollar permite considerar esta alternativa como también elegible para él.

ALTERNATIVA 4

- Sistema Operativo Windows.
- Tipo de diseño modular.
- Lenguaje de programación: Java EE con un servidor Apache Tomcat.
- Gestor de Base de Datos: Oracle.
- Se trata de una alternativa confeccionada teniendo en cuenta a posibilidad de implementar el sistema como una aplicación web.

El uso de Oracle [23] como gestor de base de datos garantiza una solución más robusto y profesional que las anteriores al tratarse de un gestor más completo en cuanto a estabilidad, escalabilidad y soporte multiplataforma y de transacciones.

3.5 Valoración de las Alternativas

Este apartado se corresponde con la Actividad EVS 5 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

Y, aunque no se van a plasmar específicamente las tres tareas propuestas por Métrica v.3 [12] para este apartado, sí que se hará una tarea conjunta de todas ellas, valorando las distintas alternativas descritas en los apartados anteriores, desde un punto de vista genérico que contemple todo lo especificado en las mismas.

3.5.1 Estudio de la Inversión

(Métrica v.3 [12] Tarea EVS 5.1)

En esta tarea hemos utilizado la técnica Análisis Coste/Beneficio para establecer su viabilidad económica. En concreto hemos determinado los costes de cada alternativa en orden a que el cliente pudiese estimar si estaba dentro de sus límites presupuestales. No se han estimado beneficios cuantificables por cuanto lo que busca el cliente es la prestación de un servicio no cuantificable en términos monetarios.

Los costes de las diversas alternativas son los siguientes:

ALTERNATIVA 1:

Herramienta	Coste para el Cliente	Cambio a €*
Linux	Gratuito	0
Java	Gratuito	0
Eclipse	Gratuito	0
MySQL Standard Edition ^[26]	2000 \$*	1.889,2 €
TOTAL:		1.889,2 €

*En este costo solo se incurrirá en caso de ampliar el uso más allá del ámbito interno

ALTERNATIVA 2:

Herramienta	Coste para el Cliente*	Cambio a €
Windows 8.1*	119 €	119 €
C#	Gratuito	0
Visual Studio .Net	Gratuito	0
Oracle ^[24]	5800 \$	5.478,68 €
TOTAL:		5.597,68 €

*(Tipo de cambio aplicado €/\$: 1.0774 \$, \$/€: 0.9446 € el oficial a fecha 09/04/2015)

* Al tratarse de herramientas y aplicaciones populares en el mercado, normalmente estarán disponibles en el cliente, por lo que podría no necesitar incurrir en el coste de las licencias o reducirse. Con todo, se expresan en su valor actual para que le cliente pueda valorarlo conforme a sus circunstancias.

ALTERNATIVA 3

Herramienta	Coste para el Cliente*
Windows 8.1	119 €
Java	Gratuito
Netbeans	Gratuito
Microsoft Access	135 €
TOTAL:	254 €

* Al tratarse de herramientas y aplicaciones populares en el mercado, normalmente estarán disponibles en el cliente, por lo que podría no necesitar incurrir en el coste de las licencias o reducirse. Con todo, se expresan en su valor actual para que le cliente pueda valorarlo conforme a sus circunstancias.

ALTERNATIVA 4

Herramienta	Coste para el Cliente	Cambio a €
Windows 8.1	119 €	119 €
Java EE	Gratuito	0
Netbeans	Gratuito	0
Oracle ^[24]	5800 \$	5.478,68 €
TOTAL:		5.597,68 €

3.5.2 Riesgos y Planificación de las Alternativas

(Métrica v.3 [12] Tarea EVS 5.2 y Tarea EVS 5.3)

En estas tareas se ha analizado el impacto en la organización visto éste desde los puntos de vista tanto tecnológico como organizativo, analizando los riesgos que cada alternativa presenta relacionados tanto con la incertidumbre y complejidad del sistema como con las implicaciones organizativas que tendría para la organización el desarrollo e implementación del sistema con cada una de ellas.

Riesgos de la Alternativa 1

- Conocimientos del Sistema Operativo Linux.
- Dificultad para implementar interfaces con el entorno de desarrollo escogido.
- Bajo rendimiento de MySQL [25] con ficheros de *logs* muy extensos.

Riesgos de la Alternativa 2

- Eficiencia del driver de conexión con Oracle desde .NET.
- No es una alternativa multiplataforma (salvo el proyecto Mono para GNU/Linux).
- Mayor complejidad en el mantenimiento e implantación del SGBD.

Riesgos de la Alternativa 3

- Rendimiento de la BBDD Access [22] con ficheros de *logs* muy extensos.
- Alternativa no escalable: Access [22] no es una BBDD distribuida.
- Netbeans [16] no es tan robusto ni utilizado a nivel profesional como Eclipse [15].

Riesgos de la Alternativa 4

- Necesidad de tener un servidor/contenedor de aplicaciones levantado.
- Complejidad del desarrollo de una aplicación web con J2EE, que seguramente exigiría el uso de frameworks adicionales: Spring, Struts, JSF, etc.
- Seguridad: al tratarse de una aplicación web se debería implementar algún mecanismo de *login* y/o seguridad para que no pueda acceder cualquiera que conozca su URL.

3.6 Selección de la Solución

Este apartado se corresponde con la Actividad EVS 6 del proceso de Estudio de Viabilidad del Sistema (EVS) de Métrica v.3 [12].

En la tarea EVS 6.1 Convocatoria de Presentación hemos puesto a disposición del cliente el presente estudio de viabilidad con el fin de que pudiese estudiar previamente su contenido, haciéndole posteriormente la presentación correspondiente en la que se aclararon y modificaron diversos puntos de la misma.

En la tarea EVS 6.2 Evaluación de las Alternativas y Selección el cliente valoró las ventajas e inconvenientes de cada una de ellas tanto desde el punto de vista económico, como tecnológico y organizativo.

En la tarea EVS 6.3 Aprobación de la Solución el cliente aprobó la alternativa 4 como la solución a implementar por encontrarlo dentro de sus posibilidades presupuestarias (motivo económico) y por ser tecnológica y operativamente viable, cumpliendo adecuadamente con los requisitos planteados en su inicio en plazos razonables y cobertura suficiente.

SOLUCIÓN A IMPLEMENTAR

- Sistema Operativo Windows
- Tipo de diseño modular
- Lenguaje de programación: C# en entorno VisualStudio.NET.
- Gestor de Base de Datos: Oracle.

Como se muestra en esta ilustración, tomada de Métrica v.3 [12], en este capítulo, partiendo de la definición del sistema realizado en el Estudio de Viabilidad (correspondiente con la actividad ASI 1), realizamos la serie de análisis que consideramos aplicables a nuestro caso, terminando con la aprobación de este análisis por parte del cliente (ASI 11).

Dado que Métrica v.3 [12] es una metodología que cubre tanto desarrollos estructurados como orientados a objetos, y aunque nuestro sistema a implementar tendrá un desarrollo orientado a objetos, hemos utilizado para nuestro análisis las actividades de ambas aproximaciones que nos han parecido más pertinentes, interpretándolas libremente para un mejor acomodo a nuestras pretensiones.

En la primera actividad, Definición del Sistema (ASI 1), se lleva a cabo la descripción inicial del sistema de información, a partir de los productos generados en el proceso Estudio de Viabilidad del Sistema (EVS). Se confirma el alcance del sistema, el catálogo de requisitos generales, los componentes tecnológicos de la solución propuesta, y se identifican los usuarios que participan en el proceso de análisis, determinando sus perfiles.

En la segunda actividad, Establecimiento de Requisitos (ASI 2), hemos elaborado un catálogo de requisitos detallado, que permita describir con precisión el sistema de información propuesto, y que además sirva de base para comprobar que es completa la especificación de los modelos obtenidos en las actividades siguientes Identificación de Subsistemas de Análisis (ASI 3), Análisis de Casos de Uso (ASI 4), Análisis de Clases (ASI 5), Elaboración del Modelo de Datos (ASI 6), Elaboración del Modelo de Procesos (ASI 7) y Definición de Interfaces de Usuario (ASI 8).

En la actividad Identificación de Subsistemas de Análisis (ASI 3), hemos estructurado el sistema de información en subsistemas de análisis, para facilitar la especificación de los distintos modelos y la traza de requisitos.

En paralelo, hemos generado el modelo que sirve de base para el diseño, partiendo con la actividad Análisis de Casos de Uso (ASI 4) y concluyendo con la actividad Elaboración del modelo de Datos (ASI 6). Es decir, combinando actividades que en principio Métrica v.3 [12] reserva para tipos de análisis diferente (análisis estructurado y análisis orientado a objetos) pero que hemos considerado adecuadas para nuestro caso concreto.

En la actividad 8 se especifican todas las interfaces entre el sistema y el usuario, tales como formatos de pantallas, diálogos, formatos de informes y formularios de entrada.

Y finalmente, en la actividad Aprobación del Análisis del Sistema de Información (ASI 11) se recibe la conformidad del Cliente para iniciar el proceso de Diseño sobre la base del Análisis del Sistema de Información presentado y aprobado.

4.1 Definición del Sistema

Este apartado se corresponde con la Actividad ASI 1 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

Las tareas que Métrica v.3 [12] propone para esta actividad, señalan que tiene como objetivo principal efectuar una descripción del sistema propuesto, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos, adecuando lo ya desarrollado en el Estudio de Viabilidad del Sistema del capítulo anterior (EVS).

En nuestro caso, consideramos el desarrollo realizado en ese Estudio, suficiente como punto de partida para definir el sistema de información propuesto, remitiéndonos directamente a él para completar este apartado.

En la tarea ASI 1.1 Determinación del Alcance del Sistema se confirma el proyecto como independiente con respecto al Plan de Sistemas de Información de la entidad, afectando únicamente a los usuarios de la aplicación, es decir, a los que la utilicen para analizar ficheros de *logs*. Para ellos, la aplicación deberá permitirles seleccionar el fichero *log* que deseen analizar y seleccionar dentro de él los aspectos que considere importantes y relevantes para sacar un informe respecto a ellos.

En la tarea ASI 1.2 Identificación del Entorno Tecnológico se confirman los elementos que configuran la solución elegida

- Sistema Operativo Windows
- Tipo de diseño modular
- Lenguaje de programación: C# en entorno VisualStudio.NET.
- Gestor de Base de Datos: Oracle.

En la tarea ASI 1.3 Especificación de Estándares y Normas se confirma, desde el punto de vista legal, la necesidad de cumplir con los requisitos legales en cuando a los datos que pueden ser de carácter personal y, la importancia de tener presente en el análisis, que se trata de ficheros de gran cantidad de registros con datos de distinta naturaleza.

En la tarea ASI 1.4 Identificación de los Usuarios Participantes y Finales, dada la importancia que la colaboración de los usuarios tiene en el proceso de obtención de los requisitos, se confirma la distinción propuesta en el Estudio de Viabilidad entre usuario Administrador y usuario Normal.

4.1.1 Marco Regulador

El presente proyecto se desarrollará bajo el marco regulador de la Ley de Protección de Datos de carácter personal, conforme a la Ley Orgánica 15/1999 [13], de manera que cualquier dato de índole privado que se encuentre será tratado con el fin de preservar la privacidad de los usuarios del sistema.

4.2 Establecimiento de Requisitos

Este apartado se corresponde con la Actividad ASI 2 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

Como segunda actividad hemos llevado a cabo la definición, análisis y validación de los requisitos a partir de la información facilitada por el usuario, completando el catálogo de requisitos obtenido en la actividad ASI 1 Definición del Sistema, tal como lo indica Métrica v.3 [12].

Como muy acertadamente señala Métrica v.3 [12], esta actividad, como en general todas las actividades del análisis, se descompone en un conjunto de tareas que, si bien tienen un orden, exige continuas realimentaciones y solapamientos. Lo importante, para nosotros, ha sido contar al final de este proceso con un catálogo de requisitos más específico que el obtenido en el Estudio de Viabilidad que facilite el diseño posterior del sistema.

Para la obtención de estos requisitos hemos utilizado como técnica fundamental los casos de uso, que es una técnica de la orientación a objetos, también utilizable en el caso estructurado. Dicha técnica ofrece un diagrama simple que sirve de guía de especificación en las sesiones de trabajo con el usuario.

Contando con esta técnica como base, hemos realizado las cuatro tareas que Métrica v.3 [12] propone para esta actividad.

En la tarea ASI 2.1 Obtención de Requisitos hemos concretado el catálogo de requisitos iniciado en el Estudio de Viabilidad mediante reuniones con los tipos de usuarios finales utilizando como técnica de obtención de requisitos el modelado de casos de uso.

En la tarea ASI 2.2 Especificación de Casos de Uso se muestra el modelado de casos de uso que hemos utilizado en las reuniones con los participantes para facilitar la obtención de información así como las fichas que los describen.

En la tarea ASI 2.3 Análisis de Requisitos se muestra como el catálogo de requisitos cubre los requerimientos del usuario y no presenta inconsistencias, ambigüedades, duplicidades o escasez de información.

En la tarea ASI 2.4 Validación de Requisitos se recibe la conformidad con los requisitos formulados tanto por parte de usuarios normales como usuarios administradores.

4.2.1 Obtención de Requisitos

(Métrica v.3 [12] Tarea ASI 2.1)

En esta tarea hemos concretado el catálogo de requisitos que debe cumplir el software, iniciado en la actividad Definición del Sistema (ASI 1 del Estudio de Viabilidad), mediante reuniones a fondo con los tipos de usuarios finales, utilizando como técnica de obtención de requisitos el modelado de casos de uso.

Nos hemos fijado en particular en las posibles restricciones del entorno, tanto hardware como software, que puedan afectar al sistema de información.

Toda la información recabada sobre los requisitos la hemos volcado en las siguientes fichas descriptivas donde se define, aparte de otros datos significativos, la prioridad que hay que asignar a cada uno de estos requisitos, considerando los criterios de los usuarios acerca de las funcionalidades a cubrir.

Identificador: RS-TXY			
Nombre	Nombre del Requisito.		
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Estabilidad del requisito.		
Descripción	Descripción breve de las especificaciones del requisito.		

Tabla 19: Especificación del formato que seguirá nuestro catálogo de requisitos del sistema

La ficha es similar a la utilizada anteriormente a propósito de los requisitos de usuario pero con algunas especificaciones nuevas que se han incorporado ahora por ser propias de los requisitos del sistema.

Sus apartados son los siguientes:

- **Identificador:** será un código único que identifique cada uno de los requisitos individualmente.

Seguirá la regla de nombrado RS-TXY, donde T indicará el tipo de requisito software que estamos describiendo y X e Y conformarán un valor numérico de dos cifras que funcionará como un contador incremental para cada uno de los grupos de requisitos de sistema existentes, empezando siempre por el 01.

Los valores de T podrán ser [28]:

- **F** – para los requisitos funcionales: derivados de los requisitos de usuario de capacidad. Especifican la funcionalidad o servicios que la aplicación debe proporcionar al usuario. Indican QUÉ debe hacer el sistema.
- **NF** – para requisitos no funcionales: derivados de los requisitos de usuario de restricción. Imponen restricciones en el producto software desarrollado. Cortan transversalmente a los requisitos funcionales, indicando CÓMO debe hacer las cosas el sistema. Pueden ser a su vez de estos tipos:

Siglas	Tipo de Requisito	Descripción
DI	Disponibilidad	Cuantifican el grado de disponibilidad de la aplicación para los usuarios.
IN	Interfaz	Describen el formato con el que la aplicación se comunica con su entorno.
ME	Manejo de errores	Indican cómo debe responder la aplicación a errores de su entorno, o a errores internos.
OP	Operación	Indican cómo va a realizar el sistema las tareas para las que ha sido construido, garantizando los niveles de servicio requeridos (tiempo de respuesta, número de usuarios, memoria utilizada).
RN	Rendimiento	Especifican restricciones de tiempo que la aplicación debe cumplir.
RS	Restricción	Describen límites o condiciones sobre cómo diseñar o implementar la aplicación. Especifican condiciones impuestas en el proyecto por el cliente, el entorno, u otras circunstancias.
SG	Seguridad	Tienen como finalidad intentar garantizar la seguridad de los datos según la LOPD [13].

Tabla 20: Tipos de requisitos no funcionales contemplados

- **Nombre:** Título descriptivo del requisito que nos indicará de forma breve y concisa en qué consiste y de qué trata.
- **Claridad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica cuán claro es el requisito en sí mismo, es decir, si tiene una única interpretación, sin ningún tipo de ambigüedad.

- **Fuente:** Nos indica si el requisito ha sido originado gracias a la interacción con el cliente o si proviene de otros medios.
- **Necesidad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica la necesidad que tiene de implementarse dentro del producto final.
- **Prioridad:** Puede tomar los valores: “baja”, “media” o “alta” y nos indica la prioridad que tiene el requisito dentro del sistema, es decir, el orden que deberían ocupar en el desarrollo de los distintos módulos del sistema.
- **Verificabilidad:** Puede tomar los valores “baja”, “media” o “alta” y nos indica si es posible verificar de forma concreta si el requisito ha sido incorporado en el producto final.
- **Estabilidad:** Podrá tomar los valores “Estable” e “Inestable” y nos indica si un requisito es susceptible o no a cambios a lo largo del desarrollo del proyecto.
- **Descripción:** Este campo es totalmente variable, y su único objetivo es el de describir detalladamente el requisito en sí.

Con estas fichas hemos descrito los 28 requisitos de sistema siguientes, que dan forma definitiva a los requisitos de usuario ya establecidos en el estudio de Viabilidad:

- REQUISITOS FUNCIONALES (6)

1. RS-F01 Borrar Datos.
2. RS-F02 Importar Fichero de Comandos Iniciales.
3. RS-F03 Importar Ficheros de *Logs*.
4. RS-F04 Analizar datos.
5. RS-F05 Mostrar Resultados del Análisis.
6. RS-F06 Tipos de Informes a Realizar.

- REQUISITOS NO FUNCIONALES (22)

De DISPONIBILIDAD DEL SISTEMA (3)

1. RS-NF-DI01 Disponibilidad del sistema.
2. RS-NF-DI02 Disponibilidad de los Datos Almacenados.
3. RS-NF-DI03 Acceso al sistema.

De INTERFAZ (5)

4. RS-NF-IN01 Interfaz Gráfica.

5. RS-NF-IN02 Idioma de la interfaz.
6. RS-NF-IN03 Menú Ayuda de la Interfaz.
7. RS-NF-IN04 Interacciones con el Usuario.
8. RS-NF-IN05 Interfaz para Informe Realizado.

De SEGURIDAD (1)

9. RS-NF-SG01 Almacenamiento de datos de forma segura.

De OPERACIÓN (4)

10. RS-NF-OP01 Borrar Datos Almacenados.
11. RS-NF-OP02 Analizar Datos Almacenados.
12. RS-NF-OP03 Análisis a Realizar.
13. RS-NF-OP04 Envío de Informes.

De RENDIMIENTO (2)

14. RS-NF-RN01 Tiempo de carga.
15. RS-NF-RN02 Tiempo de Análisis.

De RESTRICCIÓN (6)

16. RS-NF-RS01 Formato Fichero Comando Iniciales.
17. RS-NF-RS02 Formato SNOOPY.
18. RS-NF-RS03 Formato Resultados Obtenidos.
19. RS-NF-RS04 Un solo Fichero *Log* a la vez.
20. RS-NF-RS05 Formato de los Informes Realizados.
21. RS-NF-RS06 Un solo usuario a la vez.

De VALIDACIÓN (1)

22. RS-NF-VA01 Formato del correo electrónico.

A continuación se muestran las fichas de los 28 requisitos con su información correspondiente:

4.2.1.1 Requisitos Funcionales del Sistema

Identificador: RS-F01			
Nombre	Borrar Datos		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema borrará todos los datos almacenados en la base de datos, manteniendo la estructura de los mismos para futuras utilizaciones.		

Tabla 21: Requisito Software Funcional 01 (RS-F01)

Identificador: RS-F02			
Nombre	Importar Fichero de Comandos Iniciales		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema permitirá al usuario importar la lista de comandos iniciales que quiere considerar como base para el análisis de la aplicación.		

Tabla 22: Requisito Software Funcional 02 (RS-F02)

Identificador: RS-F03			
Nombre	Importar Fichero de <i>Logs</i>		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial ■ Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta ■ Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	La aplicación permitirá importar un fichero de <i>logs</i> al usuario, para su posterior análisis.		

Tabla 23: Requisito Software Funcional 03 (RS-F03)

Identificador: RS-F04			
Nombre	Analizar Datos		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	El sistema analizará los datos recogidos del fichero de <i>logs</i> a analizar y almacenados en la base de datos de la aplicación.		

Tabla 24: Requisito Software Funcional 04 (RS-F04)

Identificador: RS-F05			
Nombre	Mostrar Resultados del Análisis		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	El sistema mostrará al usuario los informes obtenidos una vez terminado de realizar el análisis solicitado.		

Tabla 25: Requisito Software Funcional 05 (RS-F05)

Identificador: RS-F06			
Nombre	Tipos de Informes a Realizar		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema permitirá al usuario escoger entre al menos dos tipos distintos de informes a realizar.		

Tabla 26: Requisito Software Funcional 06 (RS-F06)

4.2.1.2 Requisitos No Funcionales del Sistema

- Requisitos de Disponibilidad

Identificador: RS-NF-DI01			
Nombre	Disponibilidad del sistema.		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	Una vez instalada la aplicación en el equipo, el sistema estará disponible para el usuario en cualquier momento del día y del año.		

Tabla 27: Requisito Software No Funcional de Disponibilidad 01 (RS- NF-DI01)

Identificador: RS-NF-DI02			
Nombre	Disponibilidad de los Datos Almacenados		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input checked="" type="checkbox"/> Inestable		
Descripción	<p>Salvo que el cliente especifique lo contrario, los datos utilizados en la última sesión de la aplicación se quedarán almacenados en la base de datos del sistema, borrándose únicamente a través del botón de la interfaz destinado para dicho fin, con la carga de un nuevo fichero de comandos iniciales, o con la importación de un nuevo fichero de <i>logs</i> a analizar.</p>		

Tabla 28: Requisito Software No Funcional de Disponibilidad 02 (RS- NF-DI02)

Identificador: RS-NF-DI03			
Nombre	Acceso al Sistema		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	<p>La aplicación estará disponible para cualquier tipo de usuario, no contemplándose ningún mecanismo de acceso o registro en la misma.</p>		

Tabla 29: Requisito Software No Funcional de Disponibilidad 03 (RS- NF-DI03)

- **Requisitos de Interfaz**

Identificador: RS-NF-IN01			
Nombre	Interfaz Gráfica		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema proporcionará al usuario una interfaz gráfica para su interacción con el mismo, diseñada a ser posible, siguiendo las 8 reglas de oro de Shneiderman [30].		

Tabla 30: Requisito Software No Funcional de Interfaz 01 (RS-NF-IN01)

Identificador: RS-NF-IN02			
Nombre	Idioma de la Interfaz		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input checked="" type="checkbox"/> Inestable
Descripción	El idioma de la interfaz será el español por defecto, pero proporcionará al usuario la opción de cambio al inglés en el caso de que sea necesario.		

Tabla 31: Requisito Software No Funcional de Interfaz 02 (RS-NF-IN02)

Identificador: RS-NF-IN03			
Nombre	Menú Ayuda de la Interfaz		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input type="checkbox"/> Estable <input checked="" type="checkbox"/> Inestable
Descripción	La interfaz podrá tener una opción de ayuda al usuario para indicarle las funcionalidades básicas del sistema y su forma de interacción con el mismo.		

Tabla 32: Requisito Software No Funcional de Interfaz 03 (RS-NF-IN03)

Identificador: RS-NF-IN04			
Nombre	Interacciones con el Usuario		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	La interfaz proporcionará al usuario un medio de interacción con el sistema intuitivo y amigable.		

Tabla 33: Requisito Software No Funcional de Interfaz 04 (RS-NF-IN04)

Identificador: RS-NF-IN05			
Nombre	Interfaz para Informe Realizado		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Una vez terminado de realizar el análisis pertinente, el sistema abrirá automáticamente una pantalla auxiliar para mostrará al usuario los informes obtenidos.		

Tabla 34: Requisito Software No Funcional de Interfaz 05 (RS-NF-IN05)

- **Requisitos de Seguridad**

Identificador: RS-NF-SG01			
Nombre	Almacenamiento de datos de forma segura		
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Todos los datos de carácter personal, y toda la información sensible que pueda tratarse en nuestro sistema, irá debidamente almacenada y tratada según la ley orgánica de protección de datos 15/1999 LOPD [13].		

Tabla 35: Requisito Software No funcional de Seguridad 01 (RS-NF-SG01)

- **Requisitos de Operación**

Identificador: RS-NF-OP01			
Nombre	Borrar Datos Almacenados		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input checked="" type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	El sistema accederá a las tablas de la base de datos y borrar toda la información contenida en ellas, manteniendo la estructura y las relaciones correspondientes para futuras importaciones. El borrado se hará en cascada, salvando restricciones de clave primaria y clave ajena, y llevará el menor tiempo posible.		

Tabla 36: Requisito Software No Funcional de Operación 01 (RS-NF-OP01)

Identificador: RS-NF-OP02			
Nombre	Analizar Datos Almacenados		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema analizará los <i>logs</i> almacenados en la base de datos, en base a la lógica proporcionada en el fichero de Comandos Iniciales, cargado igualmente en las tablas correspondientes de la base de datos del sistema, y en función del tipo de informe solicitado.		

Tabla 37: Requisito Software No Funcional de Operación 02 (RS-NF-OP02)

Identificador: RS-NF-OP03			
Nombre	Análisis a Realizar		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El sistema realizará distintos tipos de análisis en función del tipo de informe que solicite el usuario. El modo de analizar los datos para cada tipo de informe vendrá especificado en la tabla Query Reports de la base de datos y podrá ser modificados y/o aumentados por el usuario en el caso de que así lo considere preciso.		

Tabla 38: Requisito Software No Funcional de Operación 03 (RS-NF-OP03)

Identificador: RS-NF-OP04			
Nombre	Envío de Informes		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	El sistema permitirá al usuario el envío de los informes por e-mail a una dirección de correo electrónico suministrada por él mismo.		

Tabla 39: Requisito Software No Funcional de Operación 04 (RS-NF-OP04)

- **Requisitos de Rendimiento**

Identificador: RS-NF-RN01			
Nombre	Tiempo de Carga		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente ■ Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta ■ Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	Teniendo en cuenta que un fichero de <i>logs</i> suele ser un archivo bastante pesado, el tiempo de carga de los ficheros a importar puede llegar a ser un factor crítico para el sistema, por lo que el tratamiento de los datos se hará de manera que se realicen el menor número de conexiones a base de datos, asegurando así conseguir que el tiempo de carga de la información contenida en los ficheros sea el mínimo indispensable para su correcto almacenamiento en la base de datos.		

Tabla 40: Requisito Software No Funcional de Rendimiento 01 (RS-NF-RN01)

Identificador: RS-NF-RN02			
Nombre	Tiempo de Análisis		
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	Teniendo en cuenta que los datos a analizar pueden ser muchos, el tiempo que tardará la aplicación en realizar el análisis solicitado, será el mínimo indispensable para el correcto tratamiento de los datos y la elaboración de los informes pertinentes.		

Tabla 41: Requisito Software No funcional de Rendimiento 02 (RS-NF-RN02)

- Requisitos de Restricción**

Identificador: RS-NF-RS01			
Nombre	Formato Fichero Comandos Iniciales		
Claridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input type="checkbox"/> Otros
Necesidad	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será <input type="checkbox"/> Estable <input type="checkbox"/> Inestable		
Descripción	El formato del archivo de los comandos iniciales a importar por la aplicación, será un fichero Excel con la misma estructura que el original proporcionado con el sistema.		

Tabla 42: Requisito Software No Funcional de Restricción 01 (RS-NF-RS01)

Identificador: RS-NF-RS02			
Nombre	Formato SNOOPY [1]		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente ■ Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	El formato del fichero de <i>log</i> que aceptará la aplicación, será el correspondiente al capturado con la herramienta Snoopy [1], en concreto, al especificado en el manual de usuario.		

Tabla 43: Requisito Software No Funcional de Restricción 02 (RS-NF-RS02)

Identificador: RS-NF-RS03			
Nombre	Formato Resultados Obtenidos		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente ■ Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	El formato en que se mostrarán al usuario los informes obtenidos después del análisis, será PDF.		

Tabla 44: Requisito Software No Funcional de Restricción 03 (RS-NF-RS03)

Identificador: RS-NF-RS04			
Nombre	Un solo Fichero <i>Log</i> a la vez		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente ■ Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será ■ Estable <input type="checkbox"/> Inestable		
Descripción	El sistema únicamente estará capacitado inicialmente para el análisis y almacenamiento de un solo fichero <i>log</i> a la vez.		

Tabla 45: Requisito Software No Funcional de Restricción 04 (RS-NF-RS04)

Identificador: RS-NF-RS05			
Nombre	Formato de los Informes Realizados		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	El sistema proporcionará al usuario los informes realizados únicamente en formato PDF y/o Excel.		

Tabla 46: Requisito Software No Funcional de Restricción 05 (RS-NF-RS05)

Identificador: RS-NF-RS06			
Nombre	Un solo usuario a la vez		
Claridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	■ Cliente <input type="checkbox"/> Otros
Necesidad	■ Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	■ Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		■ Estable <input type="checkbox"/> Inestable
Descripción	Si bien es cierto que se puede instalar la aplicación en varios equipos a la vez, cada una de ellas ha de trabajar siempre sobre una instancia de base de datos individual, ya que de otra manera, se corromperían los datos almacenados en las mismas solapándose ambos accesos a la vez.		

Tabla 47: Requisito Software No Funcional de Restricción 06 (RS-NF-RS06)

- **Requisitos de Validación**

Identificador: RS-NF-VA01			
Nombre	Formato del correo electrónico.		
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	<input type="checkbox"/> Cliente <input checked="" type="checkbox"/> Otros
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional		
Prioridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad	Este requisito durante el desarrollo será		<input checked="" type="checkbox"/> Estable <input type="checkbox"/> Inestable
Descripción	Se validará la dirección de correo electrónico proporcionada por el usuario de acuerdo a el estándar RFC 822 ^[1] para evitar posibles errores en el envío. En caso de no ser correcta la dirección proporcionada, el sistema emitirá un mensaje de error al usuario informándole del hecho, y no enviará el mensaje hasta que se le proporcione una dirección de correo electrónico en formato correcto.		

Tabla 48: Requisito Software No Funcional de Validación 01 (RS-NF-VA01)

4.2.2 Especificación de Casos de Uso

(Métrica v.3 [12] Tarea ASI 2.2)

Esta tarea se llevará a cabo con la participación activa de los distintos usuarios del sistema en estudio.

Objetivo: especificar cada caso de uso desarrollando el escenario.

Los casos de uso son una técnica para la especificación de requisitos en el que se identifican:

- Actores.
- Casos de uso.
- Breve descripción de cada caso de uso.

Con esta técnica hemos diseñado el siguiente Diagrama, que ha sido utilizado en las reuniones con los usuarios para facilitar la obtención y aporte de información sobre las funcionalidades del sistema.

^[1] RFC 822: formato estándar Internet para cabeceras de mensajes de correo electrónico <http://www.w3.org/Protocols/rfc822/> 07

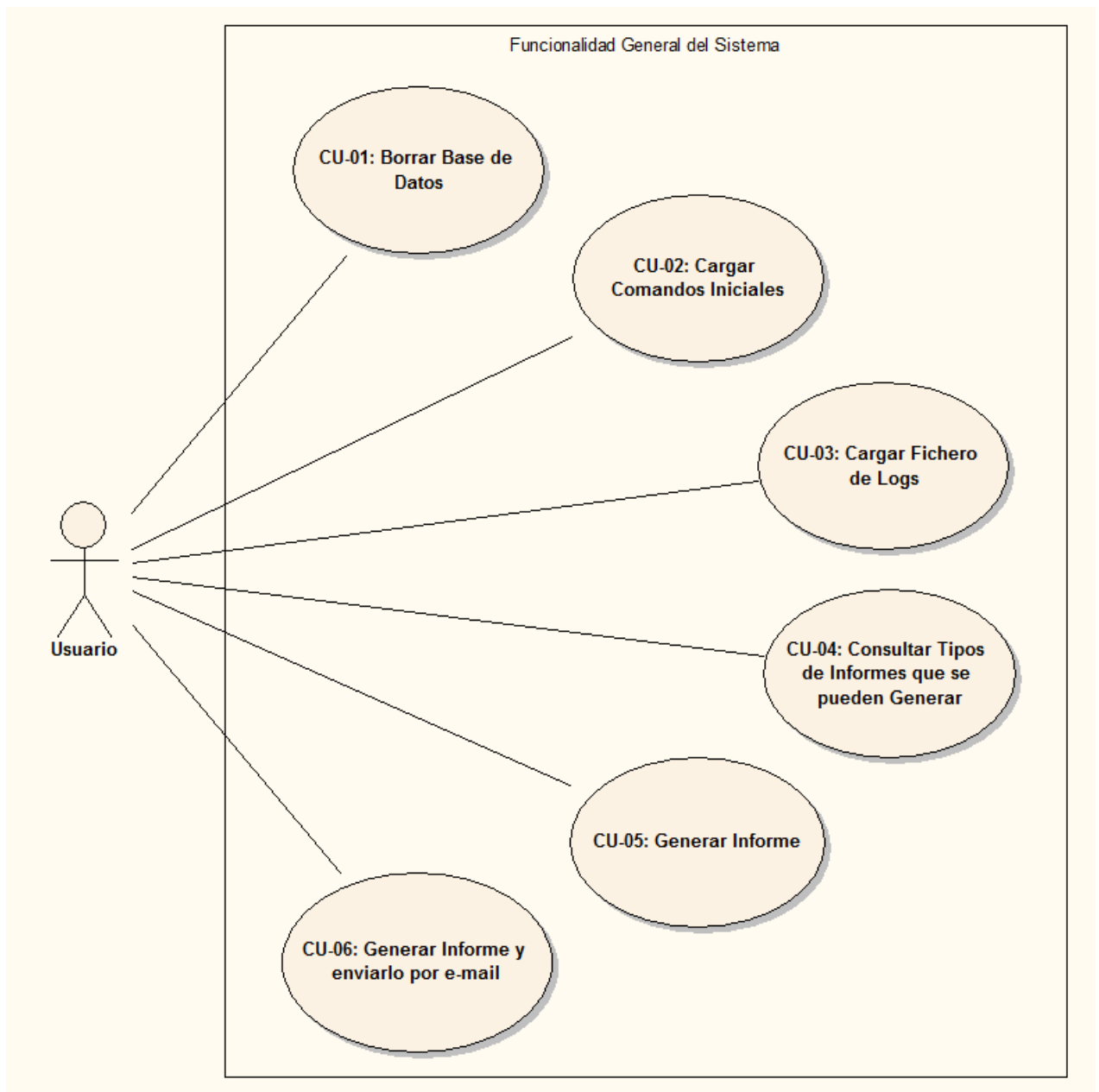


Ilustración 29: Diagrama General de Casos de Uso

Como puede verse en el diagrama se trata de los casos de usos que hará el usuario normal. No se diagramado uso especial para el usuario administrador ya que se deja a este plena libertad para analizar y acomodar la presente aplicación a los intereses que le convengan. De ahí que, pensando precisamente en él, en esta memoria se haga una descripción más detallada de la aplicación de que sería menester para el usuario común.

Para cada caso de uso se ha especificado en ficha aparte información que permite tener una idea clara del escenario de la aplicación, es decir, de la forma cómo un actor interactúa con el sistema y cuál es la respuesta que obtiene; así como las condiciones de fallo que afectan al escenario y la respuesta que da el sistema a las mismas (escenarios secundarios).

La ficha es la siguiente:

Identificador: CU-XY	
Nombre	<i>Nombre del Caso de Uso.</i>
Objetivo	<i>Descripción de lo que queremos conseguir con el Caso de Uso.</i>
Actor	Usuario Normal
Escenario	<i>Secuencia de acciones que hay que llevar a cabo para la realización del Objetivo perseguido con este Caso de Uso.</i>
Precondiciones	<i>Condiciones previas necesarias que deben cumplirse para poder realizar el Caso de Uso expuesto.</i>
Postcondiciones	<i>Condiciones que han de generarse al realizar el objetivo de este Caso de Uso.</i>
Requisitos de Usuario relacionados	<i>Requisitos de usuario del <u>apartado 3.2</u> del presente documento que se corresponde de alguna forma con la consecución de este Caso de Uso.</i>
Condiciones de fallo	<i>Condiciones de fallo que afectan negativamente al objetivo y/o al escenario de este Caso de Uso.</i>

Tabla 49: Formato para los Casos de Uso

Donde:

- El **Identificador** es un código unívoco utilizado para identificar cada uno de los requisitos que queremos especificar, siendo:
 - o CU las siglas identificativas para indicar que se trata de un Caso de Uso y,
 - o X e Y, dos cifras que representan un valor numérico con el que enumeraremos de forma incremental todos los casos de uso existentes, empezando por el 01.
- El resto de campos de la tabla se corresponden con su descripción especificada en el diseño anterior.

Mostramos a continuación las 6 fichas de casos de uso elaboradas al efecto:

Identificador: CU-01	
Nombre	Borrar Base de Datos
Objetivo	Permitir al usuario limpiar la base de datos del sistema, borrando toda la información almacenada en ella hasta ese momento.
Actor	Usuario Normal
Escenario	Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones: 1.- Abrir la aplicación; 2.- Pulsar el botón de Borrar Datos Almacenados.
Precondiciones	No hay ninguna precondición establecida, salvo la correcta conexión del sistema con la base de datos específica del mismo, y la existencia de las tablas necesarias para su correcto funcionamiento.
Postcondiciones	Una vez realizado el escenario de este caso de uso, todas las tablas de la base de datos del sistema estarán completamente vacías (menos la correspondiente a los tipos de informes a realizar).
Requisitos de Usuario	RU-CO1 RU-R05, RU-R06
Condiciones de fallo	El único fallo que podría contemplarse es que el sistema no estuviera bien conectado a la base de datos, o que las tablas necesarias para el mismo, no existieran. En el caso de que no existan datos almacenados que borrar, el sistema no dará ningún tipo de aviso o error.

Tabla 50: Caso de Uso 01 (CU-01)

Identificador: CU-02	
Nombre	Cargar Comandos Iniciales
Objetivo	Permitir al usuario cargar sus propios comandos de interés, junto con los parámetros de análisis que le interesen.
Actor	Usuario Normal
Escenario	<p>Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones:</p> <ol style="list-style-type: none"> 1.- Abrir la aplicación; 2.- Pulsar el botón de Cargar Comandos Iniciales.
Precondiciones	No hay ninguna precondición establecida, salvo la correcta conexión del sistema con la base de datos específica del mismo, y la existencia de las tablas necesarias para su correcto funcionamiento.
Postcondiciones	Una vez realizado el escenario de este caso de uso, las tablas de comandos y categorías de la base de datos del sistema contendrán únicamente la información del fichero cargado.
Requisitos de Usuario	RU-C02 RU-R05, RU-R06
Condiciones de fallo	<p>El único fallo que podría contemplarse es que el sistema no estuviera bien conectado a la base de datos, y/o que las tablas involucradas no existieran.</p> <p>En el caso de que hubiese datos almacenados anteriormente en las tablas a rellenar, el sistema no dará ningún tipo de aviso o error, eliminando todo lo almacenado anteriormente y actualizando la información con los datos leídos del fichero de comandos iniciales.</p>

Tabla 51: Caso de Uso 02 (CU-02)

Identificador: CU-03	
Nombre	Cargar Fichero de <i>Logs</i>
Objetivo	Permitir al usuario importar el fichero de <i>logs</i> que tiene interés de analizar.
Actor	Usuario Normal
Escenario	Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones: 1.- Abrir la aplicación; 2.- Pulsar el botón de Cargar Fichero <i>Log</i> .
Precondiciones	No hay ninguna precondición establecida, salvo la correcta conexión del sistema con la base de datos específica del mismo, y la existencia de las tablas necesarias para su correcto funcionamiento.
Postcondiciones	Una vez realizado el escenario de este caso de uso, las tablas de comandos usados y usuarios de la base de datos del sistema contendrán únicamente toda la información del fichero importado.
Requisitos de Usuario	RU-C03 RU-R02, RU-R05, RU-R06
Condiciones de fallo	El único fallo que podría contemplarse es que el sistema no estuviera bien conectado a la base de datos, y/o que las tablas involucradas no existieran. En el caso de que hubiese datos almacenados anteriormente en las tablas a rellenar, el sistema no dará ningún tipo de aviso o error, eliminando todo lo almacenado anteriormente y actualizando la información con los datos leídos del fichero de <i>logs</i> importado.

Tabla 52: Caso de Uso 03 (CU-03)

Identificador: CU-04	
Nombre	Consultar Tipos de Informes que se pueden Generar
Objetivo	Permitir al usuario analizar los datos del fichero de <i>logs</i> almacenado en las bases de datos del sistema.
Actor	Usuario Normal
Escenario	<p>Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones:</p> <ol style="list-style-type: none"> 1.- Abrir la aplicación; 2.- Pulsar el botón de Analizar Datos Almacenados. <p>Los distintos tipos de informes que se pueden generar se muestran en la pantalla siguiente.</p>
Precondiciones	Para que se puedan mostrar los tipos de informes a realizar, debe haberse cargado anteriormente en la base de datos del sistema (bien en esta misma sesión, o en alguna anterior) un fichero de comandos iniciales y un fichero de comandos de <i>logs</i> para analizar.
Postcondiciones	Una vez realizado el escenario de este caso de uso, se abrirá al usuario una pantalla mostrando un recuadro con los distintos tipos de informe que se pueden generar, para que pueda seleccionar el, o los, que le interesen.
Requisitos de Usuario	RU-C04, RU-C05, RU-C06 RU-R05, RU-R06
Condiciones de fallo	<p>No se contempla ninguna condición de fallo, ya que en el caso de que faltase algún tipo de información para poder realizar el análisis del fichero, el sistema mostrará un mensaje de aviso al usuario indicando la necesidad de cargar previamente la información que le falte para su correcto funcionamiento.</p> <p>Además, si no existiese ningún tipo de informe a realizar cargado en la base de datos, se mostrará el recuadro vacío, sin dar ningún tipo de error o excepción al usuario.</p>

Tabla 53: Caso de Uso 04 (CU-04)

Identificador: CU-05	
Nombre	Generar Informe
Objetivo	Permitir al usuario generar informes de análisis sobre los datos almacenados en la base de datos del sistema, y visualizarlos por pantalla.
Actor	Usuario Normal
Escenario	<p>Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones:</p> <ol style="list-style-type: none"> 1.- Abrir la aplicación 2.- Pulsar el botón de Analizar Datos Almacenados 3.- Seleccionar el tipo/s de informe/s que desea generar 4.- Pulsar el botón de Generar Informes.
Precondiciones	<p>Haber realizado anteriormente la carga de los comandos iniciales, y la carga del fichero de <i>logs</i> que queremos analizar.</p> <p>Seleccionar al menos uno tipo de informe a generar.</p>
Postcondiciones	Una vez realizado el escenario de este caso de uso, se le proporcionará al usuario un diálogo para ver si desea guardar los resultados obtenidos y/o enviarlos por e-mail; y además, se le mostrará una copia del análisis realizado en formato PDF.
Requisitos de Usuario	<p>RU-C05, RU-C06</p> <p>RU-R03, RU-R05, RU-R06</p>
Condiciones de fallo	En el caso de que el usuario no seleccione ningún tipo de informe, el sistema le notificará la necesidad de indicar al menos uno de ellos y volverá al estado anterior, sin realizar ninguna modificación.

Tabla 54: Caso de Uso 05 (CU-05)

Identificador: CU-06	
Nombre	Generar Informe, guardar copia y mandarlo por e-mail
Objetivo	Permitir al usuario guardar en una carpeta el último informe generado y/o enviarlo directamente a una dirección de correo electrónico que él mismo especifique.
Actor	Usuario Normal
Escenario	<p>Para poder llevar a cabo este caso de uso, han de realizarse las siguientes acciones:</p> <ol style="list-style-type: none"> 1.- Abrir la aplicación 2.- Pulsar el botón de Analizar Datos Almacenados 3.- Seleccionar el tipo/s de informe/s que desea generar 4.- Generar el Informe 5.- Indicar que sí que quiere guardar y/o enviar el informe por e-mail. 6.- Especificar el formato (o los formatos) en que desea obtener los informes, la dirección de correo electrónico donde quiere mandar el informe 7.- Pulsar el botón de Obtener Informe. 8.- Seleccionar la carpeta donde quiere guardar la copia.
Precondiciones	<p>Tener los comandos iniciales, y el fichero de <i>logs</i> que queremos analizar, cargados en la base de datos.</p> <p>Seleccionar al menos uno tipo de informe a generar y su formato.</p> <p>Seleccionar la opción de enviar informe por e-mail y especificar una dirección de correo válida para el destino.</p>
Postcondiciones	Una vez realizado el escenario de este caso de uso, se le abrirá al usuario una pantalla mostrando el análisis realizado en formato PDF y se le notificará el resultado del envío y almacenamiento del mismo.
Requisitos de Usuario	<p>RU-C05, RU-C06</p> <p>RU-R03, RU-R04, RU-R05, RU-R06</p>
Condiciones de fallo	En el caso de no especificarse una dirección de correo electrónico correcta para el envío de los informes y/o una carpeta destino válida, el sistema notificará al usuario dicha incorrección, indicándole la necesidad de especificar una dirección de e-mail válida para tal fin y/o una carpeta destino correcta.

Tabla 55: Caso de Uso 06 (CU-06)

4.2.3 Análisis de Requisitos

(Métrica v.3 [12] Tarea ASI 2.3)

Esta tarea consiste en el análisis de los requisitos expuestos en el presente documento, es decir, en el estudio de toda la información recogida en los catálogos de los mismos de manera que comprobemos que efectivamente no existen ningún tipo de inconsistencia ni ambigüedad en ellos, que la información no es escasa ni está duplicada y que se cumplen todas las especificaciones marcadas por el usuario.

Para ello hemos utilizado la técnica de las "Matrices de Trazabilidad" para confrontar

- Requisitos de usuario y casos de uso.
- Requisitos de usuario y requisitos funcionales.
- Requisitos de usuario y requisitos no funcionales.

Para que el cliente pueda corroborar que se cumplen todas las pautas establecidas.

Hemos confrontado también

- requisitos funcionales y requisitos no funcionales.

Para comprobar que no hay inconsistencias, ambigüedades, duplicidades o escasez de información.

Los requisitos de usuario que se utilizan como término constante de comparación son los que especifican en la tarea EVS 3.2: Identificación de Requisitos (del Estudio de Viabilidad del Sistema). Es decir:

REQUISITOS FUNCIONALES

1. RU-C01 Borrar la información almacenada en la base de datos.
2. RU-C02 Cargar los comandos iniciales.
3. RU- C03 Importar el ficho de *logs*.
4. RU-C04 Analizar el fichero de *logs*.
5. RU-C05 Visualizar los resultados del Análisis.
6. RU-C06 Realizar informes con los resultados.

REQUISITOS NO FUNCIONALES DE RESTRICCIÓN

1. RU-R01 El acceso será libre (no habrá control de acceso).
2. RU-R02 Se podrá analizar todo fichero de *log* capturado con la herramienta Snoopy.
3. RU-R03 Los informes se mostrarán en formato PDF con opción de una copia en formato PDF y/o Excel.
4. RU-R04 Se podrán enviar los informes obtenidos por e-mail.
5. RU-R05 Dispondrá de una interfaz gráfica que facilite la interacción al usuario.
6. RU-R06 Permitirá usar como lenguaje de interacción el Español y/o el Inglés.

Los casos de uso y requisitos formales y no formales con los que se comparan son los que se muestran en la Tarea ASI 2.1: Obtención de Requisitos (de este Análisis del Sistema de información).

El trabajo con estas matrices nos ha permitido, además, identificar funcionalidades o comportamientos comunes, reestructurando la información de los casos de uso a través de las generalizaciones y relaciones entre ellos.

4.2.3.1 Matrices de Trazabilidad

- **Requisitos de Usuario / Casos de Uso**

Objetivo: comprobar que todos los casos de uso aportan algo a la satisfacción de los requisitos establecidos por el usuario y que éstos están reflejados en algún caso de uso.

	Primary Use Cases::CU-01:	Primary Use Cases::CU-02:	Primary Use Cases::CU-03:	Primary Use Cases::CU-04:	Primary Use Cases::CU-05:	Primary Use Cases::CU-06:
User requirements::RU-C01...	X					
User requirements::RU-C02...		X				
User requirements::RU-C03...			X			
User requirements::RU-C04...				X		
User requirements::RU-C05...				X	X	X
User requirements::RU-C06...				X	X	X
User requirements::RU-R01...						
User requirements::RU-R02...			X			
User requirements::RU-R03...					X	X
User requirements::RU-R04...						X
User requirements::RU-R05...	X	X	X	X	X	X
User requirements::RU-R06...	X	X	X	X	X	X

Tabla 56: Trazabilidad entre Requisitos de Usuario Capacidad y Restricción y Casos de Uso

En ella puede apreciarse que solo el requisito RU-R01 El acceso será libre (no habrá control de acceso), al no requerir ninguna acción por parte del usuario, figura como no atendido por ningún caso de uso.

- **Requisitos de Usuario / Requisitos Funcionales del Sistema**

Objetivo: comprobar que todos los requisitos funcionales solicitados por el usuario han sido atendidos al redactar los requisitos funcionales del sistema.

	Functional Requirements::RS-F01:	Functional Requirements::RS-F02:	Functional Requirements::RS-F03:	Functional Requirements::RS-F04:	Functional Requirements::RS-F05:	Functional Requirements::RS-F06:
User requirements::RU-C01:...	X					
User requirements::RU-C02:...		X				
User requirements::RU-C03:...			X			
User requirements::RU-C04:...				X		
User requirements::RU-C05:...					X	
User requirements::RU-C06:...						X
User requirements::RU-R01:...						
User requirements::RU-R02:...			X			
User requirements::RU-R03:...						
User requirements::RU-R04:...						
User requirements::RU-R05:...						
User requirements::RU-R06:...						

Tabla 57: Trazabilidad entre Requisitos de Usuario y Requisitos Funcionales del Sistema

En ella puede apreciarse como todos los requisitos funcionales se corresponden con al menos un requisito de usuario de capacidad, comprobando así la correcta especificación de los requisitos establecidos en el catálogo.

- **Requisitos de Usuario / Requisitos No Funcionales del Sistema**

Objetivo: comprobar que todos los requisitos solicitados por el usuario (requisitos de capacidad y requisitos de restricción) han sido atendidos completamente al redactar los requisitos NO funcionales del sistema.

	User requirements::RU-C01:	User requirements::RU-C02:	User requirements::RU-C03:	User requirements::RU-C04:	User requirements::RU-C05:	User requirements::RU-C06:	User requirements::RU-R01:	User requirements::RU-R02:	User requirements::RU-R03:	User requirements::RU-R04:	User requirements::RU-R05:	User requirements::RU-R06:
Non-Functional Requirements::RS-NF-DI01: ...							X					
Non-Functional Requirements::RS-NF-DI02: ...				X								
Non-Functional Requirements::RS-NF-DI03: ...							X					
Non-Functional Requirements::RS-NF-IN01: ...	X	X	X	X	X	X					X	X
Non-Functional Requirements::RS-NF-IN02: ...											X	X
Non-Functional Requirements::RS-NF-IN03: ...											X	X
Non-Functional Requirements::RS-NF-IN04: ...											X	
Non-Functional Requirements::RS-NF-IN05: ...					X						X	
Non-Functional Requirements::RS-NF-OP01:...	X											
Non-Functional Requirements::RS-NF-OP02:...				X								
Non-Functional Requirements::RS-NF-OP03:...				X		X						
Non-Functional Requirements::RS-NF-OP04:...									X	X		
Non-Functional Requirements::RS-NF-RN01:...		X	X									
Non-Functional Requirements::RS-NF-RN02:...				X								
Non-Functional Requirements::RS-NF-RS01: ...		X										
Non-Functional Requirements::RS-NF-RS02: ...			X					X				
Non-Functional Requirements::RS-NF-RS03: ...					X				X			
Non-Functional Requirements::RS-NF-RS04: ...			X	X								
Non-Functional Requirements::RS-NF-RS05: ...									X	X		
Non-Functional Requirements::RS-NF-RS06: ...							X					
Non-Functional Requirements::RS-NF-VA01:...										X		

Tabla 58: Trazabilidad entre Requisitos de Usuario y Requisitos No Funcionales del Sistema

- **Requisitos Funcionales / Requisitos No Funcionales del Sistema**

Objetivo: comprobar que los requisitos NO funcionales del sistema se corresponden en forma adecuada con sus requisitos funcionales, evitando inconsistencias, redundancias o aspectos no suficientemente atendidos.

	Functional Requirements::RS-F01	Functional Requirements::RS-F02	Functional Requirements::RS-F03	Functional Requirements::RS-F04	Functional Requirements::RS-F05	Functional Requirements::RS-F06
Non-Functional Requirements::RS-NF-DI01: ...	X	X	X	X		
Non-Functional Requirements::RS-NF-DI02: ...	X			X		
Non-Functional Requirements::RS-NF-DI03: ...						
Non-Functional Requirements::RS-NF-IN01: ...	X	X	X	X	X	X
Non-Functional Requirements::RS-NF-IN02: ...						
Non-Functional Requirements::RS-NF-IN03: ...						
Non-Functional Requirements::RS-NF-IN04: ...	X	X	X	X	X	X
Non-Functional Requirements::RS-NF-IN05: ...					X	
Non-Functional Requirements::RS-NF-OP01:...	X					
Non-Functional Requirements::RS-NF-OP02:...				X		X
Non-Functional Requirements::RS-NF-OP03:...						X
Non-Functional Requirements::RS-NF-OP04:...						
Non-Functional Requirements::RS-NF-RN01:...		X	X			
Non-Functional Requirements::RS-NF-RN02:...				X		
Non-Functional Requirements::RS-NF-RS01: ...		X				
Non-Functional Requirements::RS-NF-RS02: ...			X			
Non-Functional Requirements::RS-NF-RS03: ...					X	
Non-Functional Requirements::RS-NF-RS04: ...			X	X		
Non-Functional Requirements::RS-NF-RS05: ...						X
Non-Functional Requirements::RS-NF-RS06: ...				X		
Non-Functional Requirements::RS-NF-VA01:...						

Tabla 59: Trazabilidad entre Requisitos Funcionales y No Funcionales del Sistema

4.2.4 Validación de Requisitos

(Métrica v.3 [12] Tarea ASI 2.4)

Analizada toda la información recogida en las tareas anteriores de la presente actividad, los usuarios del sistema confirman que los requisitos especificados en ellas son válidos, consistentes y completos; por lo que nos pueden servir de base para la realización del diseño del sistema.

4.3 Identificación de Subsistemas de Análisis

Este apartado se corresponde con la Actividad ASI 3 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

En esta actividad iniciamos la descomposición técnica del sistema en varios subsistemas para facilitar la traza de requisitos.

Tal como lo indica Métrica v.3 [12] esta actividad la realizamos en paralelo con el resto de actividades, ya que tal descomposición solo tiene utilidad para facilitar precisamente esas actividades y solo se justifica hacerla si realmente se necesita.

Durante el proceso de Análisis del Sistema de Información mantuvimos, por consiguiente, una realimentación y ajuste continuo con respecto a la definición de los subsistemas, sus dependencias y sus interfaces. Realizando de una manera u otra las dos tareas que Métrica v.3 [12] señala para esta actividad.

En la tarea ASI 3.1 Determinación de Subsistemas de Análisis hemos concretado los módulos que compondrán el sistema y las funcionalidades concretas que tendrán en el mismo.

En la tarea ASI 3.2 Integración de Subsistemas de Análisis hemos coordinado los módulos determinados y definido la interfaz que garantiza el funcionamiento del sistema en general.

4.3.1 Determinación de Subsistemas de Análisis

(Métrica v.3 [12] Tarea ASI 3.1)

De acuerdo al Estudio de viabilidad del sistema realizado la solución a desarrollar debía de ser modular, es decir, aislar las funcionalidades que se consideran oportunas para diseñar los pertinentes módulos capaces de realizarlas en forma independiente.

Esto está en consonancia con esta tarea de Métrica v.3 [12] en la que se dice textualmente que la descomposición del sistema en subsistemas debe estar, principalmente, orientada a los procesos de negocio, aunque también sea posible adoptar otros criterios lógicos.

El diseño modular consiste precisamente en eso, en la repartición de las funcionalidades del sistema en varios subsistemas que en conjunto aseguran el correcto funcionamiento del mismo.

Esta descomposición reduce la complejidad de implementación del sistema y facilita su mantenimiento y mejora, fomentando el desarrollo en paralelo y el aislamiento de posibles errores y fallos en el desarrollo y funcionamiento del mismo.

Para ello, y dado que nuestro análisis va orientado a objetos, hemos identificado y definido tanto los subsistemas (unidades con funcionalidad concreta) que hemos necesitado como la interfaz que permita delimitar el comportamiento de los subsistemas y su utilización en el modelo general del sistema.

Finalmente hemos asignado los requisitos y casos de uso a cada uno de los subsistemas identificados, actualizando el catálogo de requisitos con esta información.

4.3.2 Integración de Subsistemas de Análisis

(Métrica v.3 [12] Tarea ASI 3.1)

En esta tarea, cuyo objetivo según Métrica v.3 [12] es la coordinación en la elaboración de los distintos modelos de análisis de cada subsistema, hemos seguido el criterio característico del diseño modular que indica: “máxima cohesión y mínimo acoplamiento” [29] de forma que hemos acoplado en un mismo módulo las funcionalidades similares o relacionadas entre sí, y desacoplado el resto.

El objetivo de esta tarea es la coordinación asegurando la ausencia de duplicidad de elementos y la precisión en la utilización de los términos del glosario.

Esta tarea se realiza en paralelo con el resto de las actividades de elaboración de modelos del análisis, y permite tener una visión global y unificada de los distintos modelos.

El resultado ha sido el siguiente:

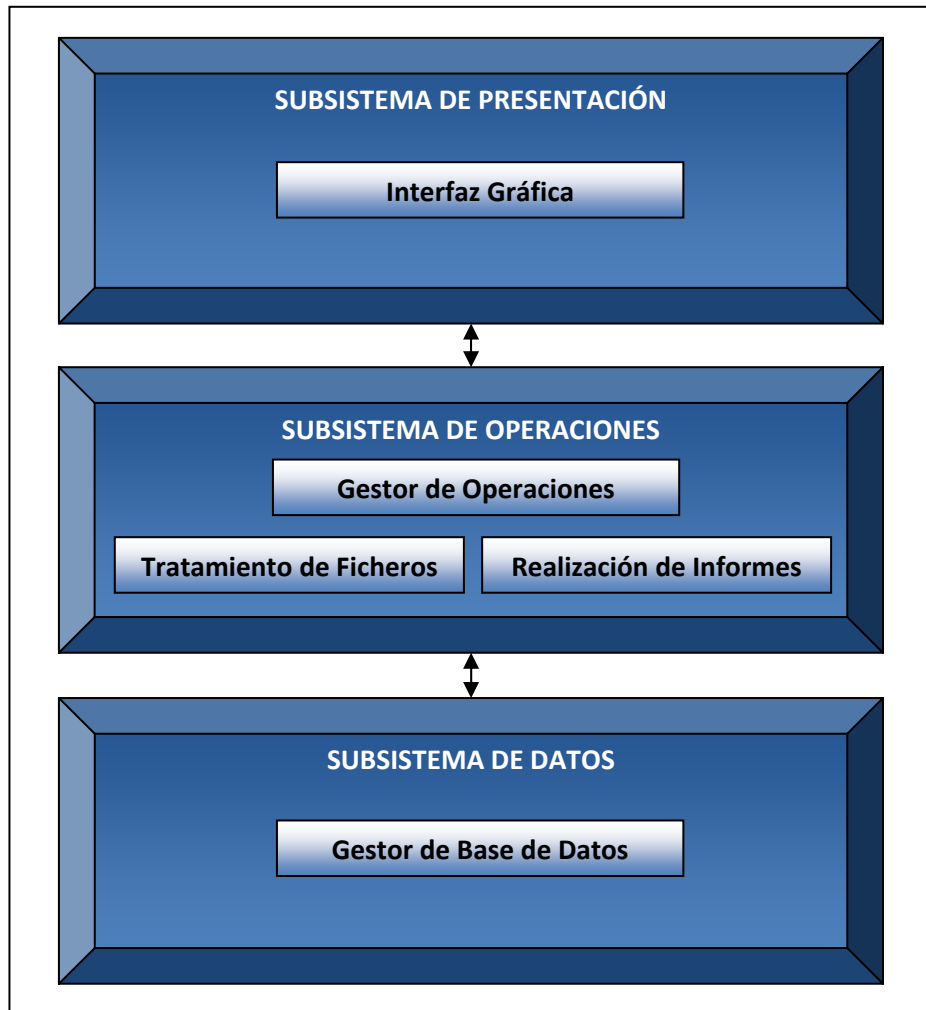


Ilustración 30: Esquema de los Subsistemas de Análisis contemplados

Donde:

- Subsistema de Presentación: se corresponde con la funcionalidad encargada de mostrar al usuario las operaciones que puede realizar con el sistema y, los resultados obtenidos de su interacción con el mismo.
 - Interfaz Gráfica: contiene la funcionalidad encargada de permitir la interacción del usuario con el sistema.
- Subsistema de Operaciones: es el encargado de gestionar todas las operaciones que realiza el sistema, permitiendo que se cumplan todas las especificaciones dadas por el cliente. Se encargará de controlar y relacionar todo el flujo de datos proveniente del Subsistema Superior con el Inferior.
 - Gestor de Operaciones: es el módulo encargado de gestionar todas las operaciones solicitadas por el usuario.
 - Tratamiento de Ficheros: este módulo contendrá todas las clases necesarias para el tratamiento de los ficheros que interactúen con la aplicación.

- Realización de Informes: este módulo será el encargado de contener la lógica necesaria para redactar y mostrar al usuario todos los tipos de informes ofrecidos por la aplicación.
- Subsistema de Datos: contiene al módulo gestor de base de datos, por lo que su funcionalidad principal es la de gestionar todo el flujo de datos proveniente de los módulos superiores, y transmitirlo directamente a acciones de inserción y/o recuperación sobre la base de datos.
 - Gestor de Base de Datos: es el módulo encargado de gestionar todas las operaciones que se realicen sobre la base de datos.

4.4 Análisis de los Casos de Uso

Este apartado se corresponde con la Actividad ASI 4 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

El objetivo de esta actividad, que Métrica v.3 [12] describe como la apropiada para el Análisis Orientado a Objetos, es identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción de dichos objetos.

Para ello realizamos en un principio las dos tareas señaladas en Métrica v.3 [12] para esta actividad.

En la tarea ASI 4.1 Identificación de Clases Asociadas a un Caso de Uso identificamos las clases cuyos objetos son necesarios para realizar un caso de uso.

En la tarea ASI 4.2 Descripción de la Interacción de Objetos describimos el comportamiento de cada caso de uso mediante la interacción de dichos objetos.

Al realizar estas tareas encontramos, como Métrica v.3 [12] contempla explícitamente, que algunos de estos objetos representan mejor la información del sistema si se les identifica como atributos en vez de como clases. Cosa que nos llevó a considerar primero la conveniencia de utilizar para nuestro análisis el modelo de datos (propio del Análisis Estructurado) como complemento o sustitución del modelo de clases (propio del Análisis Orientado a Objetos), que es lo que en definitiva hemos hecho.

4.5 Elaboración del Modelo de Datos

Este apartado se corresponde con la Actividad ASI 6 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

Como ha quedado señalado a propósito de la actividad anterior hemos considerado pertinente completar nuestro análisis realizando parcialmente esta actividad que Métrica v.3 [12] reserva para el Análisis Estructurado.

Lo hacemos porque, aunque nuestro sistema sigue un diseño orientado a objetos, el hecho de que no todas las clases de nuestro modelo impliquen almacenamiento de datos, hemos considerado que el diseño de la base de datos nos sería más fácil y confiable utilizando un modelo entidad/relación a partir de nuestro diagrama de clases que muestre con un diagrama los datos que debemos manejar en nuestro sistema.

Esta decisión está en consonancia con el objetivo que, según Métrica v.3 [12], tiene esta actividad: identificar las necesidades de información de cada uno de los procesos que conforman el sistema de información, con el fin de obtener un modelo de datos que contemple todas las entidades, relaciones, atributos y reglas de negocio necesarias para dar respuesta a dichas necesidades.

El modelo de datos que presentamos se ha realizado siguiendo un enfoque descendente (*top-down*) con las relaciones y las entidades que se han ido extrayendo principalmente en los catálogos de requisitos obtenidos anteriormente.

Para ello hemos realizado dos de las cuatro tareas señaladas en Métrica v.3 [12] para esta actividad.

En la tarea ASI 6.1 Elaboración del Modelo Conceptual de Datos hemos identificado y definido las entidades del sistema, sus atributos y relaciones.

En la tarea ASI 6.4 Especificación de Necesidades de Migración de Datos y Carga Inicial hemos descrito las necesidades de migración y carga inicial de los datos.

4.5.1 Modelo Conceptual de Datos

(Métrica v.3 [12] Tarea ASI 6.1)

En esta tarea hemos identificado y definido las entidades de datos que se contemplan para nuestro sistema, sus atributos (diferenciando aquellos que se toman como identificadores), los dominios de los atributos y las relaciones existentes entre las entidades, indicando las cardinalidades mínimas y máximas.

Las entidades, atributos, indicadores, dominios y relaciones se muestran en el siguiente diagrama y/o en la tabla que mostramos a continuación de él.

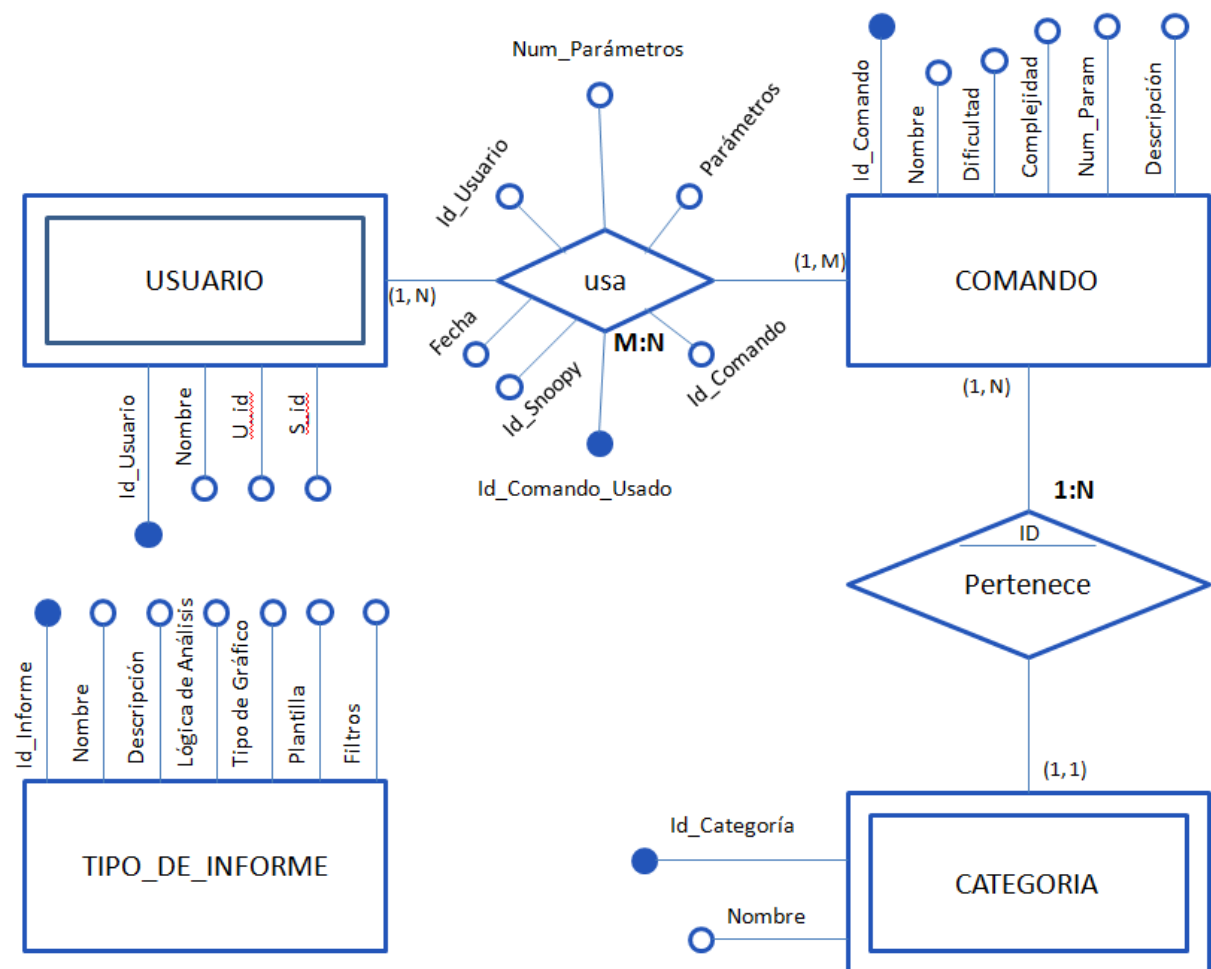


Ilustración 31: Modelo Conceptual de Datos (Modelo Entidad/Relación)

El diagrama anterior muestra las entidades de datos que se contemplan para nuestro sistema, junto con sus atributos y la identificación de las claves primarias para cada una de ellas. Faltando únicamente para definir las completamente, la especificación del dominio sobre el que se concreta cada uno de estos atributos.

Entidad	Descripción	Atributos	Dominio
USUARIO	Representa los usuarios que aparecen en el fichero de <i>logs</i> que vamos a analizar.	Id_Usuario	Entero de 8 dígitos
		Nombre	Cadena de 100 caracteres
		U_id	Entero de 8 dígitos
		S_id	Entero de 8 dígitos
COMANDO	Representa los comandos a tener en cuenta como base para el análisis de todos los datos contenidos en el fichero de <i>logs</i> que se quiere analizar.	Id_Comando	Entero de 8 dígitos
		Nombre	Cadena de 100 caracteres
		Dificultad	Entero de 4 dígitos
		Complejidad	Entero de 8 dígitos
		Num_Param	Entero de 4 dígitos
		Descripción	Cadena de 300 caracteres
CATEGORÍA	Representa la categoría a la que pertenece el comando.	Id_Categoría	Entero de 4 dígitos
		Nombre	Cadena de 100 caracteres
TIPO DE INFORME	Representa los distintos tipos de informe que podrán realizarse utilizando la aplicación de análisis propuesta.	Id_Informe	Entero de 4 dígitos
		Nombre	Cadena de 10 caracteres
		Descripción	Cadena de 300 caracteres
		Lógica de Análisis	Cadena de 100 caracteres
		Tipo de Gráfico	Entero de 2 dígitos
		Plantilla	Entero de 2 dígitos
		Filtros	Entero de 1 dígito

Tabla 60: Tabla con los dominios de los atributos del modelo de datos

La entidad CATEGORIA se ha definido como una entidad débil, puesto que depende en existencia de la entidad regular COMANDO, y sólo se contemplarán en el sistema categorías que tengan algún comando asociado. Lo mismo ocurre con la entidad USUARIO, contemplándose únicamente aquellos registros de usuarios que utilicen algún COMANDO.

En el esquema también aparecen descritas las relaciones o asociaciones existentes entre las distintas entidades del sistema.

- La interrelación USA identifica los registros de uso de los comandos parte de los usuarios, y la interrelación PERTENECE relaciona cada comando con su categoría correspondiente.

El tipo de correspondencia y la cardinalidad de cada una de estas interrelaciones también aparecen especificados en el modelo anteriormente expuesto.

La entidad que representa los tipos de informes que va a poder realizar el sistema, no se encuentra directamente relacionada con ninguna de las otras entidades del sistema, pero se ha decidido introducir aquí puesto que sí que va a ser una entidad relevante en nuestro sistema, ya que va a ser quien lleve la lógica de análisis para la realización de los distintos tipos de informes que va a proporcionar la aplicación.

El atributo *Lógica de Análisis* de la entidad *Tipo de informe*, contendrá el nombre del PL de Oracle [23] que contenga la lógica del análisis que queremos realizar para el informe asociado.

4.5.2 Especificación de Necesidades de Migración de Datos y Carga Inicial

(Métrica v.3 [12] Tarea ASI 6.4)

Se ha decidido realizar esta tarea porque, aunque no hay ningún sistema previo del que migrar los datos, sí que existe la necesidad de hacer una carga inicial de información para el correcto funcionamiento del sistema en estudio.

Por ello, a continuación haremos una primera especificación del plan de carga inicial del sistema, que se completará en el proceso Diseño del Sistema de Información (DSI), detallando algunos aspectos importantes que habrá que tener en cuenta a la hora de realizar la carga inicial de los datos:

- Planificación de la carga inicial.

Para el correcto funcionamiento del sistema, se precisa la carga inicial de dos ficheros de datos:

- Fichero de comandos iniciales
- Fichero de *logs* a analizar

La carga inicial de los datos se hará siempre que se utilice la aplicación por primera vez contra una base de datos nueva, es decir, contra una base de datos donde no se hayan cargado nunca.

El fichero de comandos iniciales ha de cargarse siempre, antes que el fichero de *logs*, pues de otra manera, no va a ser posible relacionar los comandos del análisis con los que aparecen en el fichero de *logs* que queremos analizar.

4.6 Definición de Interfaces de Usuario

Este apartado se corresponde con la Actividad ASI 8 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

El objetivo de esta actividad, según Métrica v.3 [12], es realizar un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quienes va dirigido.

Por consiguiente, hemos especificado en ella interfaces entre el sistema y el usuario: formatos de pantallas, diálogos, e informes; realizando las cinco tareas señaladas en Métrica v.3 [12] para esta actividad, excepto la tarea ASI 8.2 Identificación de Perfiles y Diálogos por cuanto en un análisis orientado a objetos esto se realiza ya en la tarea ASI 2.2 Especificación de los casos de uso.

En la tarea ASI 8.1 Especificación de Principios Generales de la Interfaz hemos establecido las directrices aplicables en los procesos de diseño y construcción, a fin de construir una interfaz de usuario acorde a sus necesidades, flexible, coherente, eficiente, y sencilla de utilizar.

En la tarea ASI 8.3 Especificación de Formatos Individuales de la Interfaz de Pantalla hemos definido el formato y contenido de cada una de las interfaces de pantalla

En la tarea ASI 8.4 Especificación del Comportamiento Dinámico de la Interfaz no hemos creído oportuno especificar con vistas al diseño su comportamiento dinámico por venir éste implícito en los controles y eventos que lo provocan

En la tarea ASI 8.5 Especificación de Formatos de Impresión hemos especificado los formatos y características de las salidas o entradas impresas del sistema.

4.6.1 Especificación de Principios Generales de la Interfaz

(Métrica v.3 [12] Tarea ASI 8.1)

En esta tarea especificaremos los estándares, directrices y elementos generales que tendremos en cuenta para la definición de la interfaz de usuario de nuestro sistema y los informes y formularios impresos que presentaremos al usuario.

El entorno en que se realizarán las diferentes interfaces del sistema, será el mismo que se ha utilizado para el desarrollo de la lógica de negocio de la aplicación: Visual Studio [18] con el lenguaje de programación C# [17].

Los principios de diseño que seguirá la interfaz de usuario de nuestro sistema cumplirán con los siguientes propósitos:

- Ha de ser Consistente, es decir, ha de evitar el uso de diferentes palabras y/o acciones para identificar propósitos iguales. Y viceversa, símbolos y/o terminologías iguales no deben ser utilizadas para fines distintos.
- Ha de tener una Gestión de Errores Sencilla, evitando en la medida de lo posible que el usuario pueda cometer un error grave, y en el caso de que se cometa uno, detectarlo y tratarlo de forma sencilla y comprensible para el usuario, de manera que le sea fácil manejarlo.
- Ha de proporcionar al usuario Ayuda sobre cómo utilizar la aplicación, y Documentación que le ayude a interactuar mejor con ella.
- Ha de ser Amigable, es decir, ha de dar a los usuarios el control de las acciones, siendo ellos mismos los encargados de iniciarlas en todo momento e informándoles de todas las acciones realizadas y el resultado de las mismas, permitiéndole igualmente continuar con su actividad mientras se ejecuten.
- Ha de tener Visibilidad, de forma que todos los controles estén claramente visibles para el usuario y con una buena representación y/o descripción de sus efectos.

- Ha de mostrar Coincidencia con el Mundo Real de manera que los controles y el lenguaje utilizado sea acorde a procesos similares en el mundo real.
- Ha de ser Comprensible, es decir, ha de permitir que el usuario sea capaz de aprender fácilmente su utilización, evitando que predomine el recuerdo de su utilización sobre el reconocimiento de su funcionalidad.
- Ha de tener un Diseño y una Estética Minimalista mostrando al usuario únicamente información relevante y evitando el uso de datos que no le sean necesarios.
- Ha de proporcionar al usuario Control y Libertad sobre la aplicación de manera que se le permitan salir de la aplicación o de la tarea que está realizando en caso de encontrarse en situaciones no deseadas, volver atrás, etc.

Para los informes y formularios impresos, se seguirán criterios similares a los utilizados en el diseño de la interfaz gráfica, cumpliéndose los siguientes aspectos:

- Han de ser Consistentes,
- Han de tener Visibilidad
- Han de mostrar Coincidencia con el Mundo Real,
- Han de tener un Diseño y una Estética Minimalista

En la tarea ASI 8.5 Especificación de Formatos de Impresión se especifican más detalladamente los formatos para los informes y formularios impresos de nuestro sistema.


4.6.2 Especificación de Formatos Individuales de la Interfaz de Pantalla


(Métrica v.3 [12] Tarea ASI 8.3)

El objetivo de esta tarea, según Métrica v.3 [12], es especificar cada formato individual de la interfaz de pantalla, desde el punto de vista estático. En el caso de un análisis orientado a objetos, estos formatos individuales van completando las especificaciones de los casos de uso.

En nuestro caso hemos realizado de esta tarea teniendo en cuenta el catálogo de casos de usos, el modelo de datos que hemos desarrollado en la actividad 6 y el catálogo de requisitos.

Definiremos aquellos aspectos de la interfaz que consideremos relevantes para su posterior diseño y construcción:

- La interfaz tendrá un tamaño fijo, proporcional al de la pantalla del dispositivo físico en el que se ejecute, imposibilitando así la opción de cambio de tamaño ad-hoc, pero permitiéndose maximizar y minimizar la aplicación en la pantalla con las opciones estandarizadas para dicho fin: 
- La ubicación de la interfaz será, inicialmente, centrada respecto a la pantalla donde se muestre, pero no se prohibirá la posibilidad de cambiarla.

- El idioma de la interfaz será por defecto el español, pero se le permitirá al usuario la opción de poder ponerlo también en inglés a través de dos símbolos universales representados por la bandera del país. Dicho cambio, supondrá únicamente la actualización de los nombres y textos al nuevo idioma, manteniéndose en todo momento las especificaciones estáticas de la interfaz.
- Todas las pantallas presentarán un menú en la barra superior que les permitirá salir del sistema y/o volver a la pantalla principal, según cuál sea el sitio donde nos encontremos.
- Dicho menú también permitirá la opción de Ayuda al usuario, proporcionándole información acerca de la aplicación y su uso.
- Si el usuario quiere salir de la aplicación de forma abrupta, todas las pantallas del sistema presentarán el botón universal de salida:  .
- El nombre del sistema aparecerá en todas las pantallas, junto con el logo.
- El formato de las pantallas será el mismo en todo momento: fuente, colores, botones, diseño, etc.
- La pantalla principal mostrará 3 botones al usuario que inicialmente estarán siempre en estado activo, pero que podrán activarse o desactivarse en función de lo que esté haciendo la aplicación.

En este apartado, puesto que se trata solamente de una tarea del proceso de Análisis, únicamente se han especificado algunos aspectos generales que tendrán de forma estática todas nuestras pantallas. Pero si se precisa de mayor información, podrán consultarse los siguientes apartados de este documento:

- [5.4 Diseño de Clases](#) (Actividad DSI 4) en concreto, el subsistema de presentación, donde veremos todos los componentes utilizados para dicho subsistema y una trazabilidad de cada uno de ellos con los requisitos analizados hasta el momento.
- Manual de usuario: aspecto final de las interfaces de usuario de la aplicación.

4.6.3 Especificación del Comportamiento Dinámico de la Interfaz

(Métrica v.3 [12] Tarea ASI 8.4)

El objetivo de esta tarea, según Métrica v.3 [12], es definir los flujos entre los distintos formatos de interfaz de pantalla, y también dentro del propio formato.

En nuestro caso el comportamiento está dirigido y representado por los controles y los eventos que provocan su activación. Al estar este comportamiento implícito en ellos, no consideramos necesario construir un modelo explícito de navegación de interfaz de pantalla pues aportaría poco al diseño posterior.

4.6.4 Especificación de Formatos de Impresión

(Métrica v.3 [12] Tarea ASI 8.5)

El objetivo de esta tarea, según Métrica v.3 [12], es especificar los formatos y características de las salidas o entradas impresas del sistema.

En esta tarea definimos los formatos individuales de informes y formularios.

Las únicas salidas impresas de nuestra aplicación son los informes, los cuales podrán tener formato Excel o PDF según lo seleccione el usuario.

Se realizarán con cada ejecución de la aplicación, de forma que no habrá ningún tipo de periodicidad marcada, sino que será a gusto y necesidad del usuario.

La confidencialidad de los informes corre igualmente por cuenta del usuario, ya que no existe ningún tipo de restricción en el acceso y uso de la aplicación. Todo aquel que tenga acceso al sistema y quiera utilizarlo, podrá hacerlo y sacar cualquier informe que esté cargado en ese momento. La única forma de evitar que otras personas puedan tener acceso a estos informes, será borrando toda la información que hemos almacenado en la base de datos después de cada utilización, de manera que nadie más que el dueño del fichero de *logs* analizado, pueda volver a reproducir los mismos resultados.

La entrega de los informes será inmediata. Una vez que el sistema haya terminado de analizar el fichero y de generar los informes solicitados por el usuario, se generará un fichero Excel con todos los informes realizados (uno por hoja) y se mostrará por pantalla un documento PDF con los resultados de los mismos. En el caso de que el usuario haya solicitado el envío de los informes por e-mail, se enviará igualmente de forma inmediata un correo con el fichero de informes en el formato/s que haya marcado el usuario, a la dirección de correo electrónico especificada.

Puesto que el sistema proporciona al usuario los resultados del análisis en un fichero impreso que puede mandarse directamente por e-mail o guardarlo en una carpeta de su ordenador que él quiera, no se contempla la necesidad de que el sistema guarde una copia de los informes realizados. Además, puesto que a igualdad de condiciones, un mismo fichero de entrada se corresponde siempre con un mismo fichero de salida, para poder volver a recuperar cualquier informe, el usuario no tiene más que volver a cargar el fichero de *logs* en la aplicación junto con el mismo fichero de comandos a analizar; obteniendo así nuevamente los resultados buscados.

Los informes cumplirán en todo momento con las especificaciones marcadas en la tarea ASI 8.1 Especificación de Principios Generales de la Interfaz, de manera que la información que mostrarán al usuario será únicamente información relevante, evitando el uso de datos que no le sean necesarios; será visible en todo momento; consistente y acorde con el mundo real.

4.7 Aprobación del Análisis del Sistema de Información

Este apartado se corresponde con la Actividad ASI 11 del proceso de Análisis del sistema de Información [ASI] de Métrica v.3 [12].

En esta actividad, que se compone de una única tarea ASI 11.1 Presentación y Aprobación del Análisis del Sistema de Información.

Hemos presentado este capítulo como documento de Análisis del Sistema de Información a nuestro cliente para su aprobación, recibiendo el visto bueno para su elaboración; por lo que lo tomaremos como documento base para el Diseño del Sistema solicitado.

5. Diseño del Sistema de Información

Se incluye el diseño de la propuesta (especificación de casos de uso, escenarios, diagramas de secuencia, esquema de base de datos, etc.)

Se realiza este diseño para definir la arquitectura del sistema y del entorno tecnológico que le va a dar soporte y, especificar de forma detallada los componentes del sistema de información.

Con toda la información que consolidamos, se generarán todas las especificaciones de construcción relativas al propio sistema; se hará una descripción técnica del plan de pruebas del mismo y; se llevará a cabo una definición de los requisitos de implantación y diseño de los procedimientos de migración y carga inicial, siempre que sean necesarios.

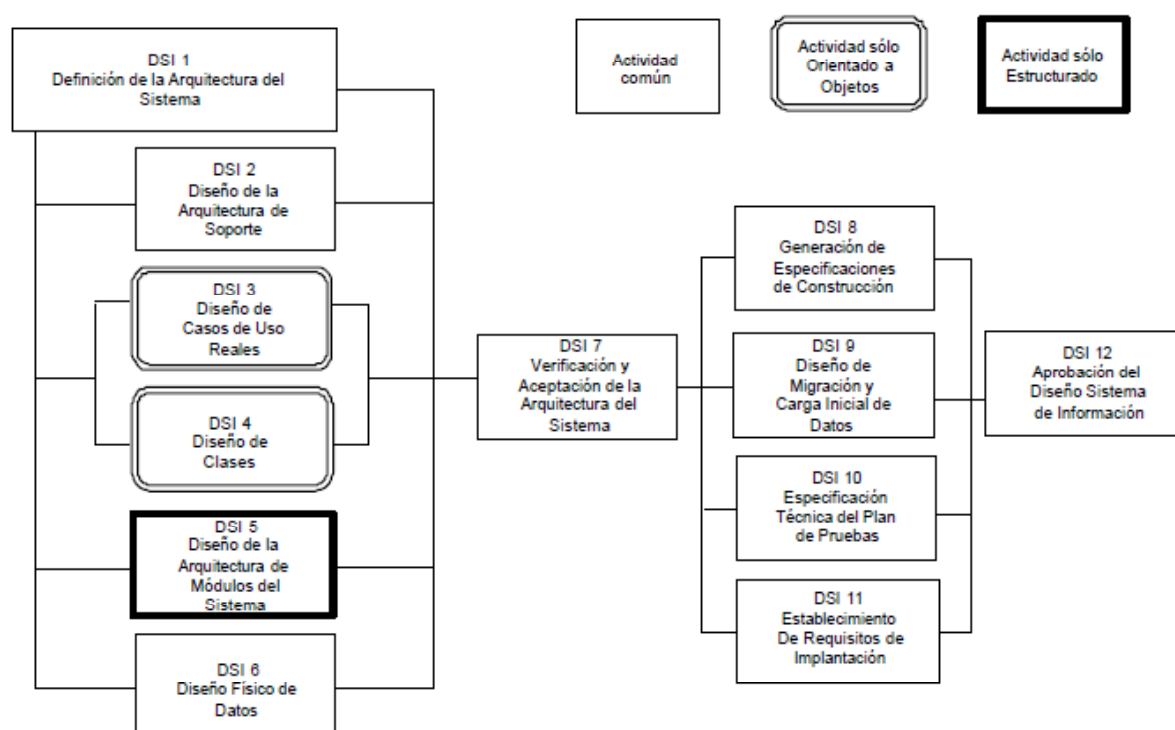


Ilustración 32: Actividades del Diseño del Sistema de Información de Métrica v.3
Fuente: Métrica v.3 [12] - Gobierno de España

Dentro de este esquema, vemos que las actividades de este proceso se agrupan en dos grandes bloques, unidos mediante una actividad de verificación y aceptación del primer bloque (DSI 7) y finalizados con otra actividad de aprobación del segundo bloque.

- Con el primer bloque se obtiene el diseño detallado del sistema de información con actividades paralelas que exigen una continua realimentación.
- Con el segundo, se complementa el diseño del sistema de información obtenido en el primero, generando todas las especificaciones necesarias para la construcción del sistema de información.

Pasemos a verlas más detalladamente.

5.1 Definición de la Arquitectura del Sistema

Este apartado se corresponde con la Actividad DSI 1 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta Actividad tiene como fin definir la arquitectura general del sistema, especificando detalladamente la infraestructura tecnológica necesaria para darle soporte y; las distintas particiones físicas del mismo (si las tiene) junto con su descomposición lógica en subsistemas de diseño y la ubicación física de cada uno de éstos subsistemas.

Además, se propone crear también un catálogo de excepciones para especificar aquellas situaciones anómalas, o secundarias, en el funcionamiento y ejecución del sistema.

Para las particiones físicas del sistema se identificarán los nodos en los que va a funcionar el sistema y las comunicaciones entre los mismos, con cierta independencia de la infraestructura tecnológica que les va a dar soporte.

Con el fin de organizar bien el diseño y facilitarlo en la medida de lo posible, se hará una división del sistema de información en subsistemas de diseño más pequeños que constituyan partes lógicas coherentes y que tengan interfaces visiblemente definidas.

Asimismo, se hará distinción entre subsistemas específicos del sistema y subsistemas específicos del soporte, con la finalidad de independizar al máximo las funcionalidades a cubrir por el sistema, de la infraestructura tecnológica que le da soporte.

Como resultado de todas estas tareas, se detallarán también los requisitos de operación, seguridad y control, derivados de las particularidades de la arquitectura del sistema propuesta, especificando igualmente los procedimientos necesarios para su correcto cumplimiento.

5.1.1 Definición de Niveles de Arquitectura

(Métrica v.3 [12] Tarea DSI 1.1)

El objetivo final de esta tarea es describir los distintos niveles de la arquitectura software de nuestro sistema describiendo los nodos que lo componen y las comunicaciones entre ellos.

Cada uno de estos nodos se corresponderá con cada una de las particiones físicas del sistema de información que estamos diseñando, teniendo cada una de ellas características propias de ejecución, de diseño y/o de construcción.

En nuestro caso, teniendo en cuenta que se recomienda identificar como nodo a todos los elementos de infraestructura más significativos de la arquitectura en la que vamos a implementar nuestro sistema, especificaremos únicamente dos nodos: uno que se encargue de la base de datos, el Gestor de Datos, y otro que se encargue de las operaciones resultantes de la interacción del usuario con el sistema, Lógica Operacional.

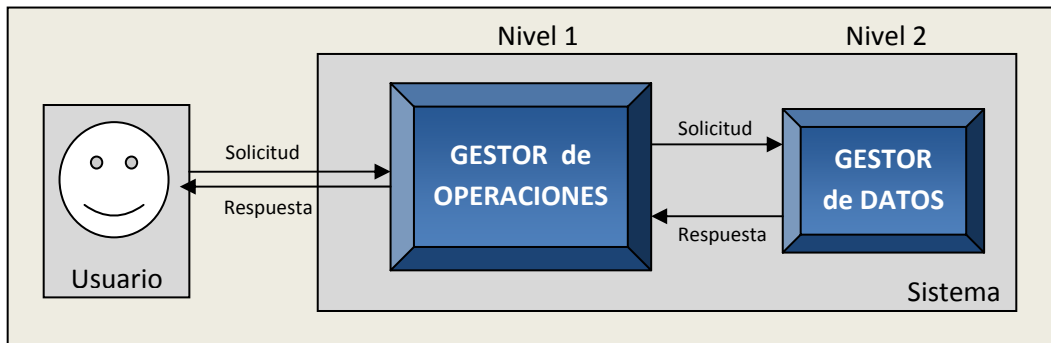


Ilustración 33: Niveles de Arquitectura

En el primer nivel de la arquitectura, tenemos el nodo encargado de proporcionar todas las funcionalidades de interacción del usuario con el sistema. Contiene toda la lógica operacional del mismo y consistirá básicamente en una aplicación pesada encargada de recibir todas las solicitudes del usuario, gestionirlas y elaborar la respuesta indicada para cada una de ellas. Dicha aplicación constará de una interfaz gráfica que proporcione al usuario una manera más amigable e intuitiva de interactuar con el sistema.

En el segundo nivel tenemos el nodo encargado de gestionar la base de datos de nuestro sistema. El sistema gestor para ello será Oracle [23], tal y como se ha especificado en el apartado 3.5 Selección de la Solución del presente documento. Las operaciones de consulta, actualización y manipulación de los datos almacenados en la base de datos, de realizarán de la forma más eficiente posible, utilizando siempre consultas lo más sencillas posibles y haciendo uso de los índices de las tablas, para agilizar las consultas de los datos contenidos en ellos. Para el proceso de inserción de datos, se ha tenido en cuenta el tratamiento de carga masiva, de manera que no se colapse el sistema y se consiga un rendimiento óptimo del mismo. Este factor es extremadamente importante en esta aplicación, ya que su función principal es el análisis de los registros almacenados en un fichero de *logs*, que en la mayoría de los casos tendrá un volumen de datos muy elevado.

Para que la aplicación funcione correctamente, es necesario que ambos nodos estén interconectados: bien a través de una red interna o externa, o bien dentro del mismo ordenador. Dicha conexión se establecerá especificando el puerto y la *ip* donde se encuentra el sistema gestor de la base de datos, en nuestro caso Oracle [23], de manera que la aplicación sea capaz de conectarse a él internamente e interactuar con la base de datos especificada para tal fin.

Si nos remontamos al Análisis del Sistema de Información realizado en capítulos anteriores, en concreto a la actividad ASI 3 de Métrica v.3 [12] plasmada en el apartado 4.2 Identificación de Subsistemas de Análisis, podríamos ubicar los subsistemas analizados, dentro de los niveles de arquitectura especificados, de la siguiente manera:

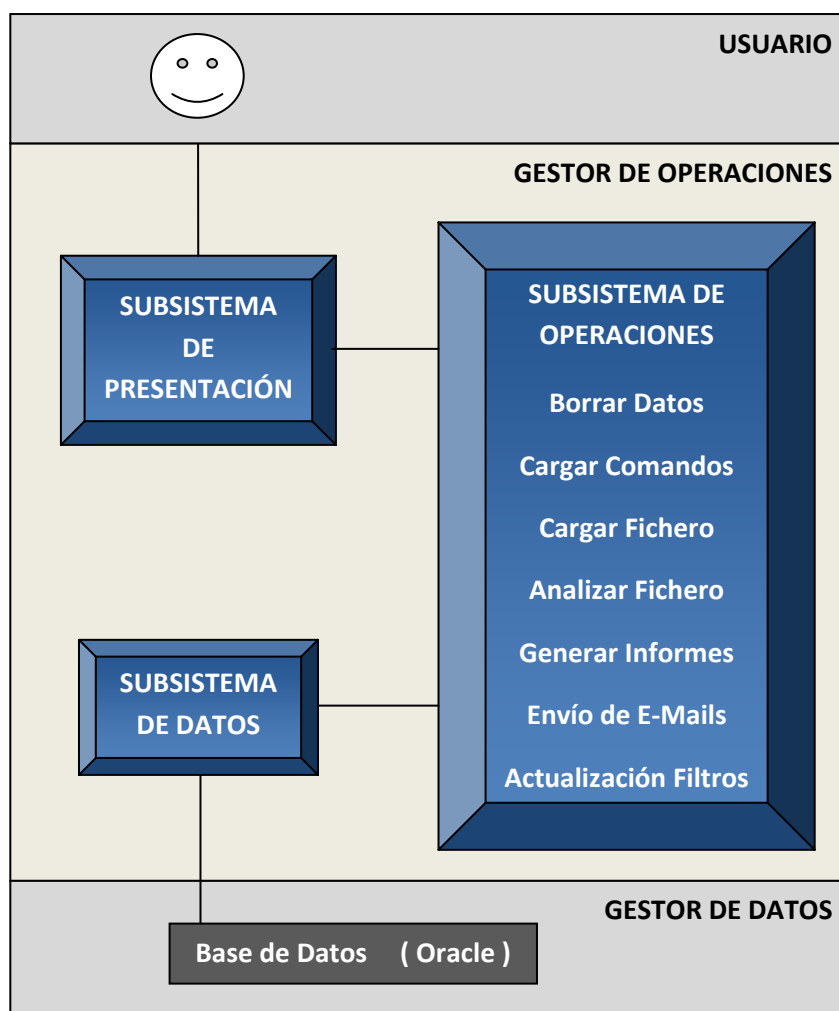


Ilustración 34: Niveles de Arquitectura con Subsistemas de Análisis

Tal y como se puede apreciar en la imagen, y en correspondencia a lo que se ha detallado anteriormente, los tres subsistemas del análisis se encuentran en el primer nivel de la arquitectura, el que contiene la lógica de negocio del sistema, quedando el segundo nivel, el inferior, para uso exclusivo de la base de datos.

La lógica de los datos se encuentra en el Subsistema de Datos, que es el módulo encargado de controlar todas las operaciones que se van a realizar sobre la base de datos del sistema; las funcionalidades de interacción entre el usuario y la aplicación se encuentran ubicadas en el Subsistema de Presentación que será el encargado de presentar al usuario las opciones de uso de la aplicación; y el Subsistema de Operaciones será el que encargado poseer la lógica operacional del sistema.

5.1.2 Especificación de Excepciones

(Métrica v.3 [12] Tarea DSI 1.3)

La finalidad de esta tarea consiste en definir los comportamientos excepcionales del sistema, es decir, aquellas situaciones que puedan producirse de forma no habitual en el mismo y que reflejen una ejecución o un funcionamiento anómalo de éste.

Para la catalogación y especificación de las excepciones, utilizaremos el siguiente formato de tabla, de manera que queden totalmente definidos su nivel de especificación, los criterios de catalogación y la clasificación de las mismas.

Identificador: EX_XY	
Nombre	<i>Nombre descriptivo de la Excepción.</i>
Tipo y/o Descripción	<i>Breve descripción de en qué consiste la excepción.</i>
Condiciones Previas	<i>Estado del sistema antes de que se produzca la excepción.</i>
Elemento/s Afectado/s	<i>Elementos del Sistema que se ven afectados por la excepción.</i>
Respuesta del Sistema	<i>Respuesta del sistema ante la excepción capturada.</i>
Elemento Asociado	<i>Elemento del sistema asociado a la respuesta esperada.</i>
Formato Respuesta	<i>Indica el formato en que se informará al usuario de la excepción capturada.</i>

Tabla 61: Especificación del formato que seguirá nuestro catálogo de excepciones

Donde, al igual que ocurría en las especificaciones de los requisitos, X e Y son dos cifras que representan un valor numérico con el que enumeraremos de forma incremental todas las excepciones contempladas en nuestro catálogo, empezando por el 01.

Las excepciones que plasmaremos en el catálogo del sistema, serán aquellas que estén relacionadas principalmente con el funcionamiento general del mismo; pero también se tendrán en cuenta aquellas excepciones particulares que se vayan identificando durante las actividades del diseño de detalle.

Identificador: EX_01	
Nombre	Error Acceso a la Base de Datos
Tipo y/o Descripción	No es posible acceder a la base de datos del sistema.
Condiciones Previas	La base de datos del sistema está caída y/o no hay conexión con el sistema. Uso de la aplicación normal, antes de un acceso a la base de datos.
Elemento/s Afectado/s	Subsistema de Datos (Capa propia del Modelo) y Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje con el siguiente texto: "Error de conexión a la Base de Datos del Sistema"
Elemento Asociado a la Respuesta	Será el Subsistema de Datos (capa Modelo) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 62: Excepción del Sistema 01 (EX_01)

Identificador: EX_02	
Nombre	Base de Datos Vacía
Tipo y/o Descripción	No hay datos almacenados en la Base de Datos
Condiciones Previas	Uso de la aplicación normal, antes de un acceso a la base de datos por cualquier motivo.
Elemento/s Afectado/s	Subsistema de Datos (Capa propia del Modelo) y Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando la necesidad de insertar información en la base de datos.
Elemento Asociado a la Respuesta	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 63: Excepción del Sistema 02 (EX_02)

Identificador: EX_03	
Nombre	Error Formato Fichero Comandos Iniciales
Tipo y/o Descripción	El fichero de comandos iniciales que se le ha pasado a la aplicación, no cumple con las reglas de formato establecidas.
Condiciones Previas	Funcionamiento totalmente normal, antes de la importación del fichero de comandos iniciales con el formato erróneo.
Elemento/s Afectado/s	Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando la necesidad de importar un fichero de comandos iniciales con el formato correcto.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 64: Excepción del Sistema 03 (EX_03)

Identificador: EX_04	
Nombre	Error Formato Fichero <i>Log</i>
Tipo y/o Descripción	El fichero de <i>logs</i> que se le ha pasado a la aplicación para analizar, no cumple con las reglas de formato establecidas.
Condiciones Previas	Funcionamiento totalmente normal, antes de la importación del fichero de <i>logs</i> a analizar con formato erróneo.
Elemento/s Afectado/s	Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando la necesidad de importar un fichero de <i>logs</i> con el formato correcto.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 65: Excepción del Sistema 04 (EX_04)

Identificador: EX_05	
Nombre	Error Envío Correo Electrónico
Tipo y/o Descripción	Se produce un error en el envío del correo electrónico y éste no puede llegar a su destino.
Condiciones Previas	Funcionamiento normal, antes de intentar enviar un mensaje por e-mail con los informes realizados.
Elemento/s Afectado/s	Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando que se ha producido un error en el envío del mensaje, y que no va a realizar la tarea.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 66: Excepción del Sistema 05 (EX_05)

Identificador: EX_06	
Nombre	Error PL Incorrecto
Tipo y/o Descripción	El PL almacenado en la tabla de informes, correspondiente al informe que ha solicitado el usuario, no está bien definido (no existe o está mal escrito).
Condiciones Previas	Funcionamiento normal, antes de realizar el informe cuyo PL está mal especificado.
Elemento/s Afectado/s	Subsistema de Datos (Capa propia del Modelo) y Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando que el PL correspondiente al informe "Nombre Informe" no ha sido reconocido.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 67: Excepción del Sistema 06 (EX_06)

Identificador: EX_07	
Nombre	Error Fichero de Informes
Tipo y/o Descripción	Se ha producido un error durante el trabajo con el fichero Excel que contiene los informes realizados.
Condiciones Previas	Funcionamiento normal antes de empezar a escribir los informes realizados en el fichero correspondiente.
Elemento/s Afectado/s	Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	Se mostrará por pantalla al usuario un mensaje indicando el error producido durante el trabajo con el fichero de informes realizados.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 68: Excepción del Sistema 07 (EX_07)

Identificador: EX_08	
Nombre	Error Tabla de Informes Vacía
Tipo y/o Descripción	La tabla que contiene todos los tipos de informes a realizar por la aplicación se encuentra vacía.
Condiciones Previas	Funcionamiento normal antes de solicitar el análisis del fichero de <i>logs</i> .
Elemento/s Afectado/s	Subsistema de Operaciones (Capa propia del Controlador)
Respuesta del Sistema	La aplicación mostrará al usuario un mensaje indicando que no hay informes cargados en la base de datos.
Elemento Asociado	Será el Subsistema de Operaciones (capa Controlador) el encargado de generar la respuesta a este evento, y el Subsistema de Presentación (capa Vista) el encargado de mostrársela al usuario.

Tabla 69: Excepción del Sistema 08 (EX_08)

5.1.3 Identificación de Subsistemas de Diseño

(Métrica v.3 [12] Tarea DSI 1.5)

Esta tarea tiene como propósito general dividir el sistema de información en subsistemas de diseño más pequeños que reduzcan su complejidad y faciliten su mantenimiento.

Esta división, se realiza tomando como referencia los subsistemas de análisis determinados en el apartado 4.2 Identificación de Subsistemas de Análisis del presente documento, y los niveles de arquitectura especificados en esta actividad.

La división se hará teniendo en cuenta aspectos como:

- La facilidad de mantenimiento del mismo,
- La reutilización de elementos del propio sistema o de su instalación,
- La optimización de recursos,
- Las características de ejecución,
- La funcionalidad común.

De esta manera, nos encontramos con que el sistema de información que estamos diseñando tendrá los siguientes subsistemas de diseño principales:

- 1) **Subsistema de Datos**: subsistema de **soporte** encargado de gestionar todo lo relacionado con la base de datos: inserción de registros, recuperación de datos, actualización, borrado, etc.
- 2) **Subsistema de Operaciones**: subsistema **específico** de la aplicación que contempla las funcionalidades propias del sistema, es decir, que se encarga de la operatividad del mismo.
- 3) **Subsistema de Presentación**: subsistema de **soporte** que se encarga de gestionar la interfaz de usuario del sistema, proporcionándole todas las operaciones de interacción con el mismo, y mostrándole todos los resultados obtenidos de su aplicación.

Estando cada uno de ellos a su vez dividido de la siguiente manera:

- 1) Subsistema de Datos

1.1) Subsistema Entidades

Es el encargado de contener toda la lógica de las entidades propias del sistema (comandos, categorías, usuarios y comandos usados – o registros *logs*) especificándose sus atributos y los métodos públicos para su obtención (*get*) y modificación (*set*) desde otras clases externas.

1.2) Subsistema de Acceso a Datos

Es el encargado de contener toda la lógica de acceso a los datos de la base de datos y, por tanto, de su operatividad: acceso para consultas, inserciones, borrados o modificaciones.

Las clases de ambos subsistemas estarán directamente relacionadas mediante la conexión directa que existe entre las entidades lógicas de nuestro sistema, y las tablas físicas de almacenamiento que van a existir en la base de datos.

2) Subsistema de Operaciones

2.1) Subsistema de Ejecución de Procesos en *Background*

Es el encargado de gestionar la ejecución de los procesos más pesados del sistema en segundo plano, evitando así el congelamiento de la interfaz del usuario durante dicha tarea.

2.2) Subsistema de Carga Inicial

Es el encargado de gestionar la carga de los comandos iniciales y la lógica del análisis desde el fichero de comandos iniciales proporcionado por el usuario. Se encargará de leer toda la información contenida en el mismo y almacenarla en las distintas tablas de almacenamiento de la base de datos destinadas para dicho fin.

2.3) Subsistema de Carga *Logs*

Es el encargado de gestionar la carga de los registros almacenados en el fichero de *logs* proporcionado por el usuario. Se encargará de leer toda la información contenida en el mismo y almacenarla en las distintas tablas de almacenamiento de la base de datos destinadas para dicho fin.

2.4) Subsistema de Generación Informes

Es el encargado de analizar la información almacenada en la base de datos del sistema y de generar los informes solicitados por el usuario en el formato especificado por él mismo. El análisis lo realizará en base a los criterios establecidos y almacenados en la tabla de informes del sistema y los mostrará según los requisitos especificados por el usuario.

2.5) Subsistema Auxiliar

Se encarga de gestionar la interacción entre el Subsistema de Operaciones y el Subsistema de Datos.

3) Subsistema de Presentación

3.1) Subsistema Principal

Es el encargado de presentar al usuario todas las interacciones iniciales con el sistema (borrar datos almacenados, cargar fichero de comandos iniciales y lógica de análisis, cargar fichero de *logs*, analizar los registros *logs* almacenados en la base de datos del sistema).

3.2) Subsistema Informes

Es el encargado de presentar al usuario todas las opciones para la configuración de los informes que quiere generar: tipo de informe y formato del mismo, solicitud de envío por e-mail, etc.

3.3) Subsistema Tarea Completada

Es el encargado de presentar al usuario las interacciones necesarias para realizar un nuevo informe sobre los datos almacenados, volver a la pantalla principal para empezar de nuevo, o salir completamente de la aplicación.

5.1.4 Especificación del Entorno Tecnológico

(Métrica v.3 [12] Tarea DSI 1.6)

Esta tarea busca definir en detalle los distintos elementos de la infraestructura técnica que soportan al sistema de información que estamos diseñando.

Para ello, definiremos bien la implementación concreta de los nodos contemplados y las comunicaciones que habrá entre ellos, tal y como se ha definido en la tarea Definición de Niveles de Arquitectura (apartado 5.1.1), y los agruparemos según los siguientes conceptos:

1. Elementos Hardware:

Puesto que lo que estamos diseñando es una aplicación pesada, contenida en sí misma, para su correcto funcionamiento y ejecución, será indispensable utilizar una máquina o equipo físico con unas características mínimas de:

- ✓ **Procesador** 1.4 GHz De CPU.
- ✓ **Memoria RAM** Con 512 MB.
- ✓ **Disco duro** de 20 GB como mínimo.
- ✓ **Red de comunicaciones:** equipo conectado a una red con salida para el envío de correos electrónicos.
- ✓ **Tarjeta de vídeo y monitor** SVGA, con una resolución de 1024x768.
- ✓ **Ratón y Teclado** compatible.

2. Elementos Software:

Además, será imprescindible que el equipo cuente con los siguientes requisitos software para ejecutar correctamente la aplicación:

- ✓ **Sistema Operativo:** Windows 7
- ✓ **Base de datos:** Oracle [23] versión 10g Express Edition
- ✓ **Framework de ejecución:** .Net Framework 3.5
- ✓ **Para la generación de Informes y documentación:** Acrobat Reader PDF y la herramienta Microsoft Excel 2003, por lo menos.

Las pautas de crecimiento que se contemplan para el sistema de información, serán básicamente aquellas relacionadas con el almacenamiento de registros *log*. Esto es, irán siempre en función del tamaño de fichero de *logs* que cargue el usuario para analizar, teniendo una media de 15.000 registros por MB de almacenamiento.

3. Comunicaciones:

Puesto que la aplicación será capaz de enviar un e-mail con los informes generados, el equipo donde se ejecute necesitará al menos una comunicación de internet con un 1 MB.

5.1.5 Especificación de Requisitos de Operación y Seguridad

(Métrica v.3 [12] Tarea DSI 1.7)

Esta tarea tiene como objetivo principal definir los procedimientos de seguridad y operación necesarios para asegurar el correcto funcionamiento del sistema, garantizar su protección y minimizar el riesgo de pérdida, alteración o, consulta indebida de la información contenida en el mismo.

Puesto que el sistema que estamos diseñando pertenece a la tipología de aplicación pesada, los requisitos de seguridad en el control de acceso al sistema y a sus recursos, recaerá directamente sobre los del propio equipo en que vaya a ser instalada; y no impone ningún tipo de requisito de seguridad adicional, ya que únicamente se encarga de realizar informes sobre el análisis de los registros contenidos en el fichero *log* que le proporcione el usuario. Por lo que será el propio fichero *log* el encargado de cumplir con las reglas de seguridad, mantenimiento de la integridad y confidencialidad de los datos contenidos en el mismo, siendo su propio gestor el encargado de realizar las copias de seguridad pertinentes, de la recuperación de los datos y su periodicidad y, de la recuperación ante catástrofes, en caso necesario.

En cuanto a los requisitos de operación para los distintos elementos del sistema, únicamente deberemos tener en cuenta un requisito de operación indispensable que hay que realizar siempre antes de utilizar el sistema por primera vez, para lograr el correcto funcionamiento del mismo; es la:

- Carga inicial de la base de datos:

Para el correcto funcionamiento del sistema, es indispensable realizar una carga inicial de las estructuras de almacenamiento de los datos, la cual se llevará a cabo siguiendo los siguientes pasos:

- Arranque del sistema gestor de base de datos elegido (en nuestro caso, Oracle [23]).
- Acceso al mismo como usuario administrador:
 - ✓ Cargar de los scripts de creación de tablas,
 - ✓ Carga de los scripts de creación de secuencias
 - ✓ Carga del script de inserción de los informes a realizar.
 - ✓ Carga de los scripts de creación de los procedimientos almacenados (PL's).

5.2 Diseño de la Arquitectura de Soporte

Este apartado se corresponde con la Actividad DSI 2 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta actividad tiene como objetivo principal especificar la arquitectura de soporte del sistema, lo que supone definir el diseño de los subsistemas de soporte identificados en la actividad anterior, y determinar los mecanismos genéricos de diseño, los cuales sirven de guía en la utilización de diferentes estilos de diseño, tanto en el ámbito global del sistema de información, como en el diseño detallado del mismo.

5.2.1 Diseño de Subsistemas de Soporte

(Métrica v.3 [12] Tarea DSI 2.1)

En esta tarea se especificarán y diseñarán los distintos módulos y/o clases, que constituyen los subsistemas de soporte que hemos especificado en la tarea de Identificación de Subsistemas de Diseño (apartado 5.1.3 del presente documento).

El diseño de nuestro sistema podría decirse que se corresponde con el patrón de diseño MODELO – VISTA – CONTROLADOR, pues separa la lógica de los datos, las funcionalidades de la interfaz y, las operaciones del sistema, quedando las correspondencias establecidas de la siguiente manera:

- 1) Subsistema de Datos → MODELO → contiene la lógica de los datos del sistema y por ende, proporciona al sistema todos los servicios necesarios para poder consultar y manipular la información almacenada en la base de datos.
- 2) Subsistema de Presentación → VISTA → es la encargada de generar las interfaces del sistema para su interacción con el usuario o con otros sistemas en futuras mejoras o actualizaciones.
- 3) Subsistema de Operaciones → CONTROLADOR → contiene la lógica operacional del sistema, implementando todas las operaciones que se pueden realizar sobre el mismo.

A continuación, para que se entienda mejor, mostraremos una figura con el esquema de la arquitectura software que estamos diseñando:

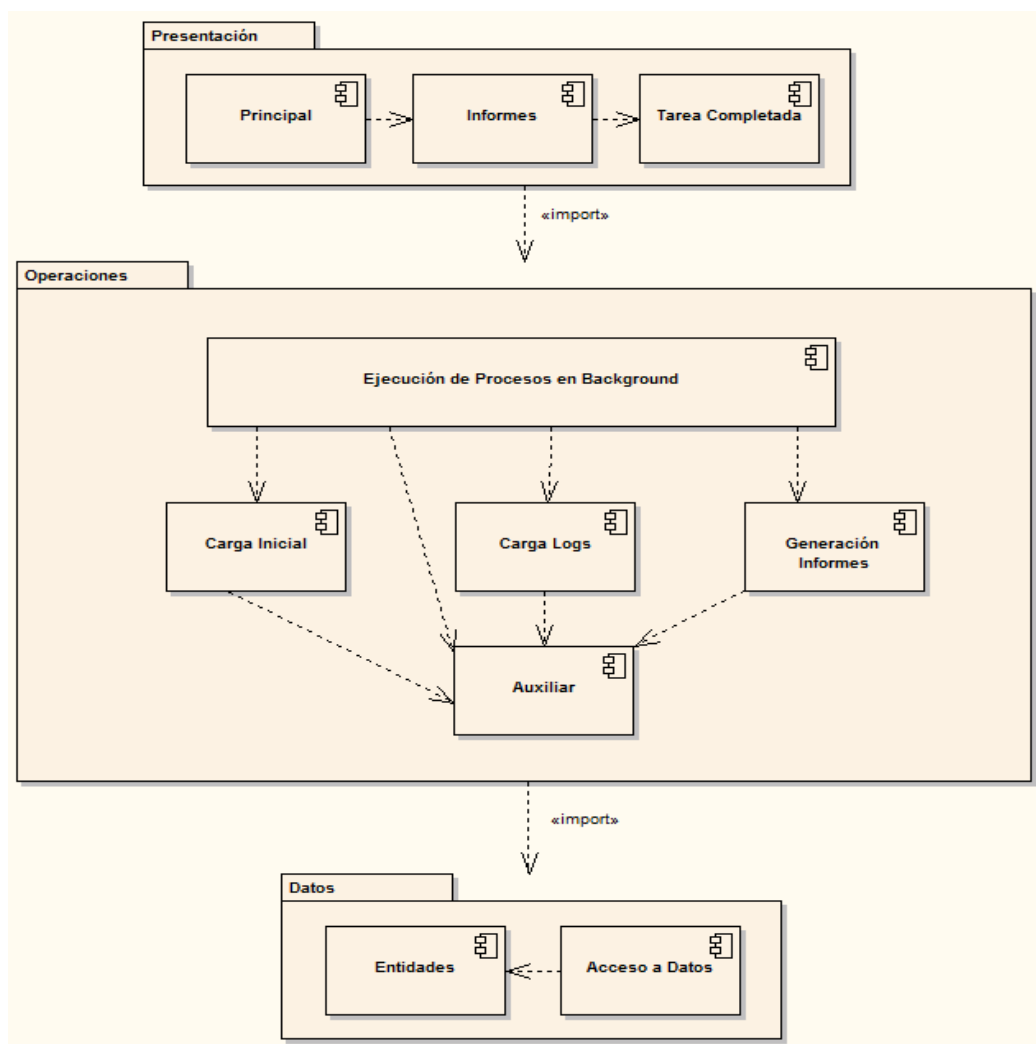


Ilustración 35: Arquitectura de Soporte del Sistema

5.3 Diseño de Casos de Uso Reales

Este apartado se corresponde con la Actividad DSI 3 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta actividad se realiza en paralelo con la actividad DSI 4 de Métrica v.3 [12] y consiste en la especificación del comportamiento del sistema de información para todos y cada uno de los casos de uso contemplados en el apartado de Especificación de Casos de Usos del presente documento. Dicha especificación se realizará indicando los objetos o subsistemas del diseño que interactúan en el mismo y las operaciones de las clases e interfaces que intervienen, incluyendo las clases de diseño que correspondan y teniendo en cuenta las restricciones del entorno tecnológico especificadas, esto es, los detalles relacionados con la implementación del sistema. Además, los escenarios de los casos de uso se elaborarán a nivel de subsistemas de esta forma que se delimiten las interfaces necesarias para cada uno de ellos.

5.3.1 Identificación de Clases Asociadas a un Caso de Uso

(Métrica v.3 [12] Tarea DSI 3.1)

Esta tarea tiene como objetivo principal identificar las clases asociadas a cada caso de uso especificado, a partir del conjunto de clases definidas paralelamente en el apartado 5.4 del presente documento (Diseño de Clases).

Para la representación de estas asociaciones, se va a seguir un formato de tabla que especifique todas las clases y subsistemas relacionados con cada caso de uso especificado en el catálogo de casos de uso del presente documento (apartado 4.1.1 Especificación de Casos de Usos).

Identificador del Caso de Uso	CU-01	
Nombre del Caso de Uso	Borrar Base de Datos	
Subsistemas Relacionados		Clases Relacionadas
Presentación. Principal		Form1.cs FormUtils.cs FormReferences.cs
Operaciones. Ejecución de en <i>Background</i>		DeleteDataWorker.cs
Operaciones. Auxiliar		ChargeData.cs ProcessResult.cs
Datos. Acceso a Datos		CategoryDatos.cs CommandDatos.cs UserDatos.cs UsedCommandDatos.cs

Tabla 70: Clases Asociadas al Caso de Uso 01 del Sistema (CU-01)

Identificador del Caso de Uso	CU-02	
Nombre del Caso de Uso	Cargar Comandos Iniciales	
Subsistemas Relacionados		Clases Relacionadas
Presentación. Principal		Form1.cs FormUtils.cs FormReferences.cs
Operaciones. Ejecución en <i>Background</i>		RestartDataBaseWorker.cs
Operaciones. Carga Inicial		InitialCharge.cs
Operaciones. Auxiliar		ChargeData.cs ProcessResult.cs
Datos. Entidades		Command.cs Category.cs
Datos. Acceso a Datos		CommandDatos.cs CategoryDatos.cs Sequences.cs

Tabla 71: Clases Asociadas al Caso de Uso 02 del Sistema (CU-02)

Identificador del Caso de Uso	CU-03	
Nombre del Caso de Uso	Cargar Fichero de <i>Logs</i>	
Subsistemas Relacionados		Clases Relacionadas
Presentación. Principal		Form1.cs FormUtils.cs FormReferences.cs
Operaciones. Ejecución en <i>Background</i>		FileTreatmentWorker.cs
Operaciones. Carga <i>Logs</i>		FileTreatment.cs
Operaciones. Auxiliar		ChargeData.cs ProcessResult.cs
Datos. Entidades		UsedCommand.cs User.cs
Datos. Acceso a Datos		UserCommandDatos.cs UserDatos.cs Sequences.cs

Tabla 72: Clases Asociadas al Caso de Uso 03 del Sistema (CU-03)

Identificador del Caso de Uso	CU-04	
Nombre del Caso de Uso	Consultar Tipos de Informes que se pueden Generar	
Subsistemas Relacionados		Clases Relacionadas
Presentación. Principal		Form1.cs FormUtils.cs FormReferences.cs
Presentación. Informes		Form2.cs
Operaciones. Auxiliar		ChargeData.cs ProcessResult.cs
Datos. Entidades		QueryReport.cs
Datos. Acceso a Datos		QueryReportDatos.cs

Tabla 73: Clases Asociadas al Caso de Uso 04 del Sistema (CU-04)

Identificador del Caso de Uso	CU-05	
Nombre del Caso de Uso	Generar Informe	
Subsistemas Relacionados		Clases Relacionadas
Presentación. Principal		Form1.cs FormUtils.cs FormReferences.cs
Presentación. Informes		Form2.cs
Presentación. Tarea Completada		Form3.cs
Operaciones. Ejecución en <i>Background</i>		GenerateReportWorker.cs UpdateReportFiltersWorker.cs
Operaciones. Generación Informes		Report.cs
Operaciones. Auxiliar		ChargeData.cs ProcessResult.cs
Datos. Entidades		QueryReport.cs ActiveFilter.cs FilterDataResult.cs
Datos. Acceso a Datos		QueryReportDatos.cs

Tabla 74: Clases Asociadas al Caso de Uso 05 del Sistema (CU-05)

Identificador del Caso de Uso	CU-06	
Nombre del Caso de Uso	Generar Informe y mandarlo por e-mail	
Subsistemas Relacionados	Clases Relacionadas	
Presentación. Principal	Form1.cs FormUtils.cs FormReferences.cs	
Presentación. Informes	Form2.cs	
Presentación. Tarea Completada	Form3.cs	
Operaciones. Ejecución en <i>Background</i>	GenerateReportWorker.cs	
Operaciones. Generación Informes	Report.cs	
Operaciones. Auxiliar	ChargeData.cs ProcessResult.cs	
Datos. Entidades	QueryReport.cs	
Datos. Acceso a Datos	QueryReportDatos.cs	

Tabla 75: Clases Asociadas al Caso de Uso 06 del Sistema (CU-06)

5.3.2 Diseño de la Realización de los Casos de Uso

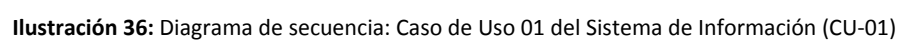
(Métrica v.3 [12] Tarea DSI 3.2)

Esta tarea tiene como objetivo definir la interacción que hay entre los distintos objetos identificados en la tarea anterior, para realizar técnicamente cada caso de uso especificado en nuestro sistema de información.

El punto de partida serán los escenarios determinados en el análisis, en concreto en el apartado de Especificación de Casos de Usos, y se detallan teniendo en cuenta el entorno tecnológico que da soporte al sistema y los mecanismos genéricos de diseño.

Para la realización de los diagramas de secuencia, se ha decidido llegar únicamente a las clases del nivel de acceso a datos, dejando sin representar los *beans* del sistema, es decir, las entidades, por considerarse que no agregan una mayor especificación de detalle y sí que complican el tamaño y la visibilidad del conjunto de los diagramas de secuencia expuestos.

- ✓ Borrar Base de Datos



- Diagrama de secuencia del Caso de Uso 02 del Sistema de Información:

✓ Cargar Comandos Iniciales

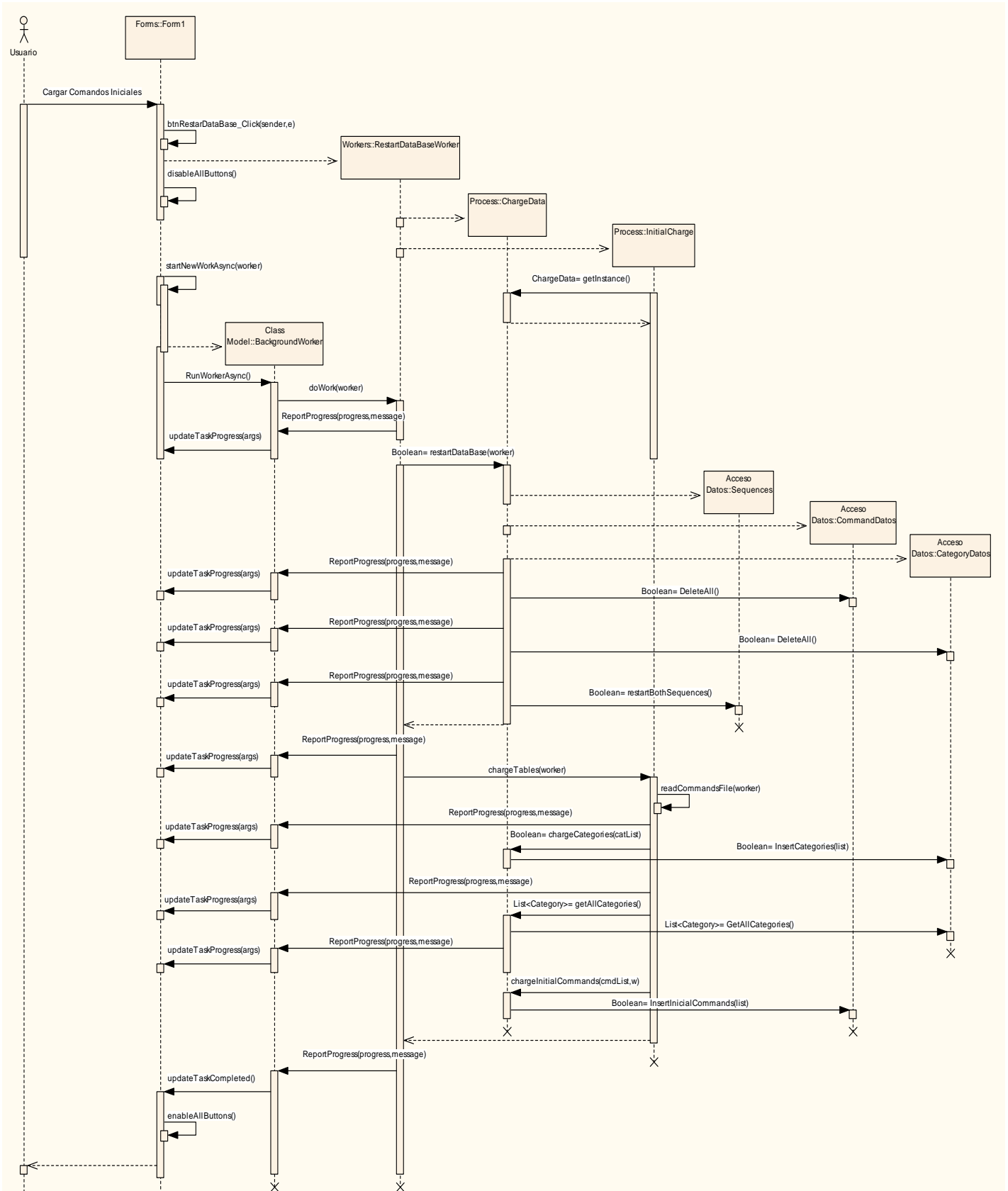
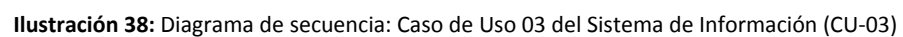


Ilustración 37: Diagrama de secuencia: Caso de Uso 02 del Sistema de Información (CU-02)

- ✓ Cargar Fichero de *Logs*



- Diagrama de secuencia del Caso de Uso 04 del Sistema de Información
 - ✓ Consultar Tipos de Informes que se pueden Generar

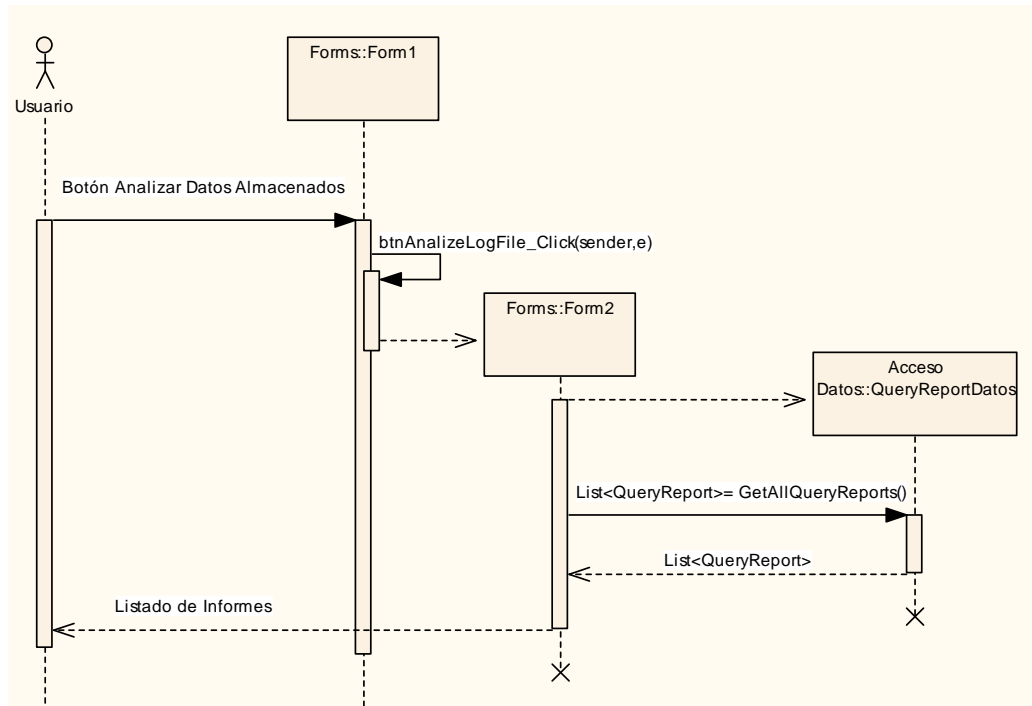


Ilustración 39: Diagrama de secuencia: Caso de Uso 04 del Sistema de Información (CU-04)

- Diagrama de secuencia del Caso de Uso 05 del Sistema de Información:

✓ Generar Informe

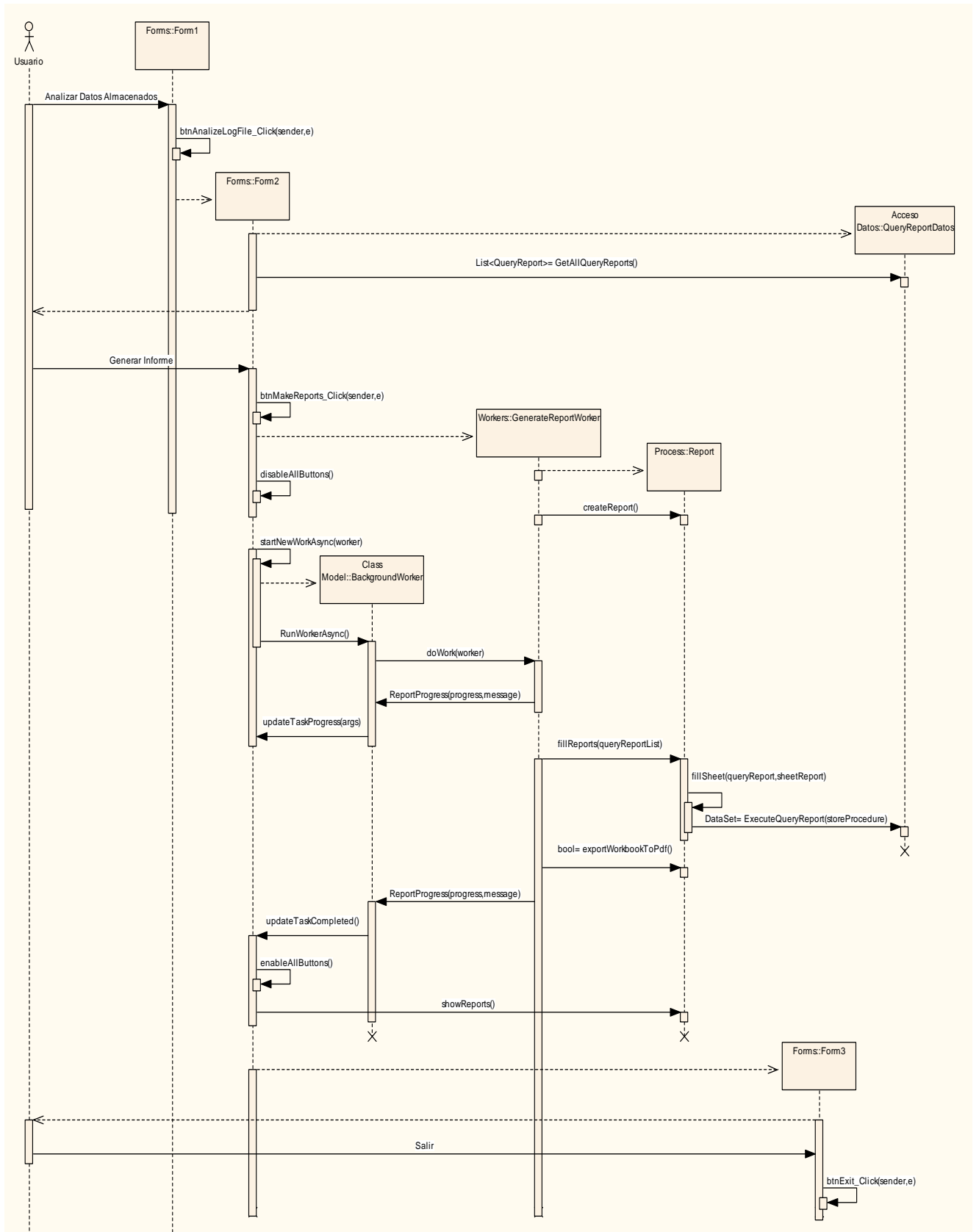


Ilustración 40: Diagrama de secuencia: Caso de Uso 05 del Sistema de Información (CU-05)

- Diagrama de secuencia del Caso de Uso 06 del Sistema de Información:

✓ Generar Informe, guardar copia y mandarlo por e-mail

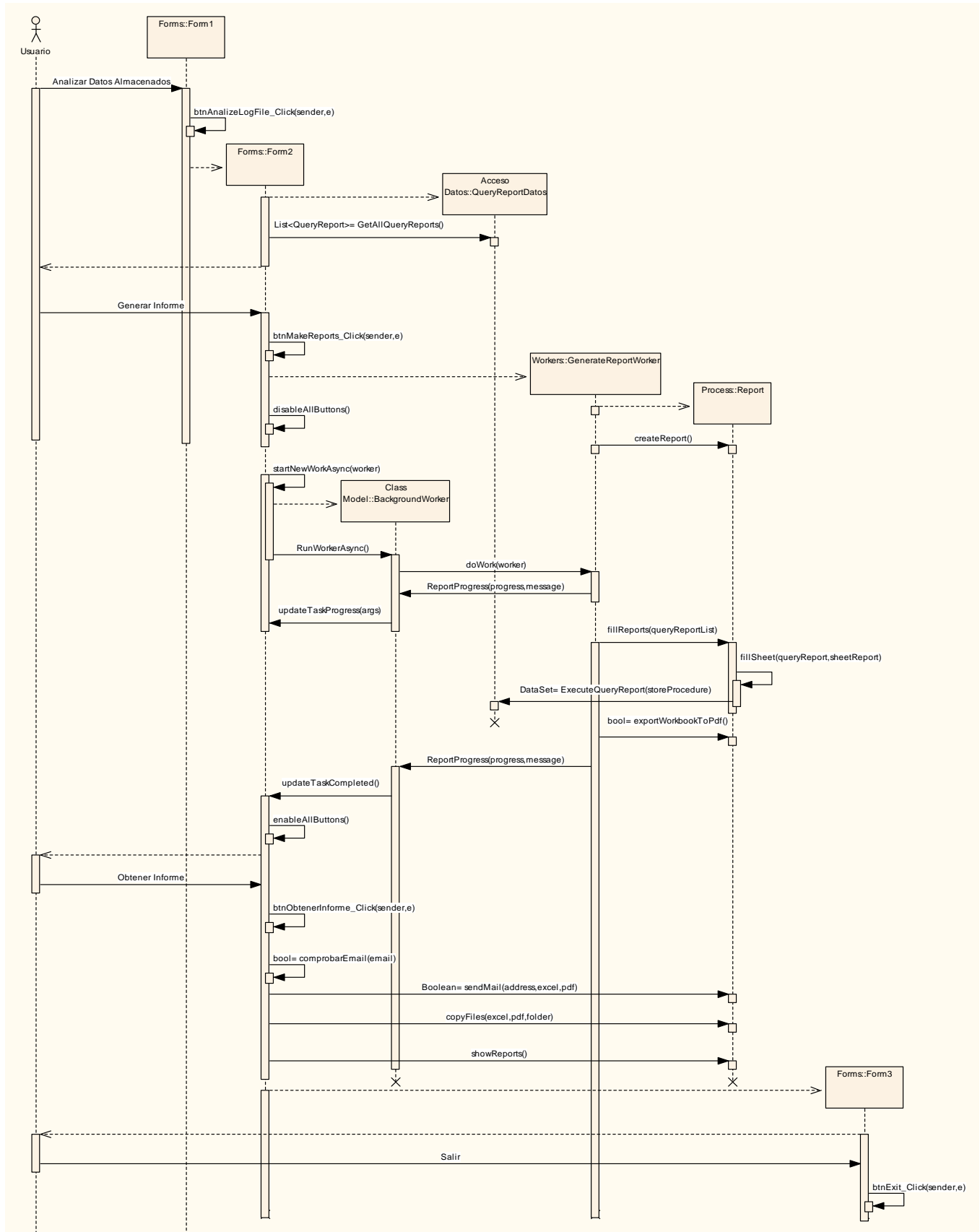


Ilustración 41: Diagrama de secuencia: Caso de Uso 06 del Sistema de Información (CU-06)

En el Diagrama de Secuencia del Caso de Uso CU-01 – Borrar Base de Datos, se puede ver cómo el usuario interactúa con la Interfaz Principal del Sistema para solicitar el borrado de los datos almacenados. Esta petición, provoca que la aplicación borre de la base de datos todos los datos almacenados anteriormente. No produciéndose ningún tipo de error en caso de que la base de datos se encuentre vacía, e informando al usuario en todo momento del progreso de la tarea y del término y resultado de la misma, cuando ésta ya se haya completado.

En el Diagrama de Secuencia del Caso de Uso CU-02 – Cargar Comandos Iniciales, nuevamente se puede ver cómo el usuario interactúa con la Interfaz Principal del Sistema para solicitar la carga del fichero de comandos iniciales. Esta petición, provoca que la aplicación solicite al usuario la selección del archivo que quiere cargar y que ejecute todos los métodos necesarios para almacenarlo de forma óptima y correcta en la base de datos. El sistema, antes de cargar toda la información del fichero, borrará los comandos y categorías existentes en la base de datos (puesto que la aplicación únicamente acepta una lógica de análisis por sesión) y, una vez borrados todos los datos, procederá al almacenamiento de la información contenida en el fichero de comandos iniciales desgranándola en los comandos y categorías propios de la lógica de análisis del sistema. Durante el proceso, la aplicación informará al usuario del progreso de la tarea y del término y resultado de la misma, cuando ésta ya se haya completado.

En el Diagrama de Secuencia del Caso de Uso CU-03 – Cargar Fichero de *Logs*, se puede ver, una vez más, cómo el usuario interactúa con la Interfaz Principal del Sistema para solicitar la carga del fichero de *logs* que quiere analizar. Esta petición, provoca que la aplicación solicite al usuario la selección del archivo que quiere cargar y que ejecute todos los métodos necesarios para almacenarlo de forma óptima y correcta en la base de datos. El sistema, antes de cargar toda la información del fichero, borrará los registros *log* (o comandos usados) y los usuarios existentes en la base de datos (puesto que la aplicación únicamente acepta la carga de un único fichero de *logs* a la vez) y, una vez borrados todos los datos, procederá al almacenamiento de la información contenida en el fichero de *logs* proporcionado por el usuario desgranándola en los comandos usados y usuarios que tendrá que analizar posteriormente el sistema. Durante el proceso, la aplicación informará al usuario del progreso de la tarea y del término y resultado de la misma, cuando ésta ya se haya completado.

El Diagrama de Secuencia del Caso de Uso CU-04 – Consultar Tipos de Informes es bastante más básico que los anteriores, puesto que esta funcionalidad del sistema, a pesar de ser una de las más importantes para la aplicación, es de las más simples en cuanto a codificación se refiere, ya que al tener una tabla con todos los tipos de informes a realizar almacenados en la base de datos del sistema, bastará con mostrar al usuario el nombre de todos los registros de la misma. Esto se hace a través de dos formularios distintos, el formulario principal (donde se solicitará el análisis de los datos almacenados) y el formulario de generación de informes, donde se mostrarán al usuario todos los tipos de informe que podrá realizar con el sistema.

Los Diagramas de Secuencia de los Casos de Uso CU-05 y CU-06 – Generar Informe y Generar Informe, guardarlo y mandarlo por e-mail, podría decirse que son dos casos de uso complementarios. En el primero, se ofrece la posibilidad al usuario de Generar un Informe de Análisis de los datos almacenados en el sistema, y visualizarlo directamente, pero sin guardarse una copia o mandarla por e-mail. Simplemente generar el informe y visualizarlo.

Mientras que en el segundo, se le amplía además la opción de elegir el formato en que desea obtener el informe, guardarse una copia del mismo y mandarla por e-mail a la dirección de correo electrónico que él mismo especifique. En ambos casos, el usuario solicita la realización de la tarea a través tanto del formulario principal, donde solicita el análisis de los datos almacenados en el sistema, como del formulario secundario, donde seleccionará el tipo de informe que desea realizar y solicitará la generación del mismo, siendo necesario además, en el segundo caso de uso, que el usuario solicite la obtención del informe realizado, el formato en que desea obtenerlo, la carpeta donde desea guardarlo y la dirección de correo electrónico donde quiere enviarlo. Durante el proceso de generación del informe, la aplicación informará al usuario del progreso de la tarea y del término y resultado de la misma, cuando ésta ya se haya completado. En el caso de la obtención del informe, al ser una tarea que se realiza de forma casi inmediata, no se ha visto la necesidad de informar al usuario del progreso de la tarea, sino, simplemente de la consecución de la misma y de los resultados obtenidos.

5.4 Diseño de Clases

Este apartado se corresponde con la Actividad DSI 4 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

El objetivo principal de esta actividad, que se realiza en paralelo con la anterior, es generar el modelo de clases de diseño de nuestro sistema.

Dicho modelo recogerá de forma detallada la especificación de cada una de las clases diseño, en función del estudio de los escenarios de Casos de Uso especificados en la actividad anterior, indicando sus atributos, operaciones, métodos, la visibilidad de cada uno de ellos con respecto al resto de las clases del modelo y, la jerarquía establecida entre ellas.

5.4.1 Identificación de Clases Adicionales

(Métrica v.3 [12] Tarea DSI 4.1)

En esta tarea se llevará a cabo la identificación del conjunto de clases que componen el modelo de clases de diseño de nuestro sistema de información – denominadas en Métrica v.3 [12] “*Clases Adicionales*” – teniendo en cuenta siempre que éstas pueden ser construidas por el propio equipo de desarrollo, cogidas en forma de bibliotecas, facilitadas por el entorno tecnológico o facilitadas por el propio entorno de trabajo.

Puesto que el diseño de nuestro sistema se corresponde con el patrón de diseño MODELO – VISTA – CONTROLADOR, que separa la lógica de los datos, las funcionalidades de la interfaz y, las operaciones del sistema, el diseño de las clases se realizará acorde a dicho patrón, quedando la vista general del sistema de la siguiente manera:

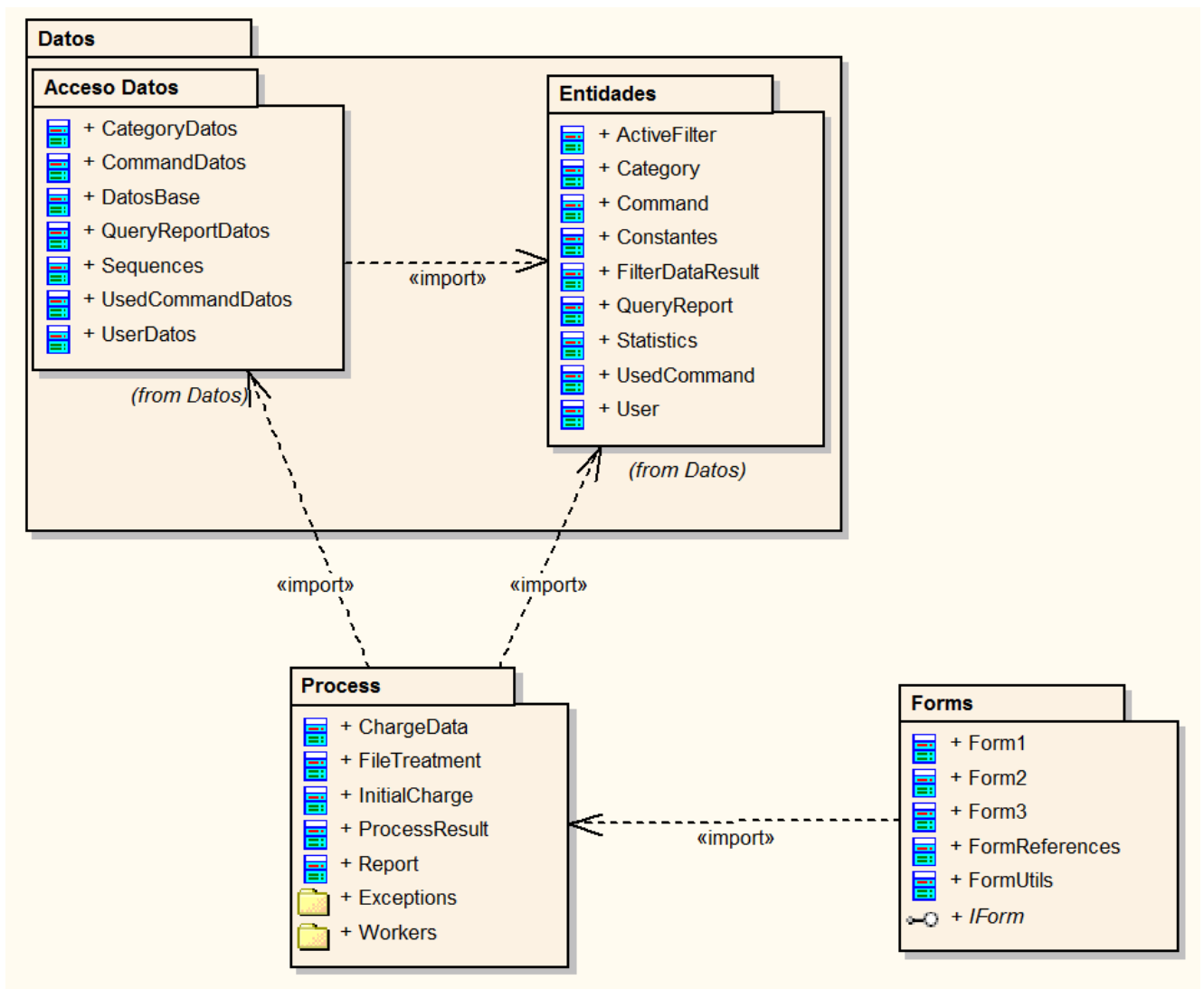


Ilustración 42: Diagrama de Clases General del Sistema de Información

5.4.2 Identificación de los Atributos y Métodos de las Clases

(Métrica v.3 [12] Tarea DSI 4.3 y 4.4)

En este apartado, identificaremos los atributos y los métodos de las clases especificadas en la tarea anterior, indicando su tipo (en el caso de los atributos) y sus parámetros (en el caso de los métodos), pero sin ahondar más en el detalle de su descripción, puesto que todo el código de la aplicación estará comentado correctamente en origen.

Al final del diagrama de clases de cada componente, insertaremos una tabla descriptiva de cada una de las clases involucradas en el mismo, de manera que se pueda seguir la trazabilidad de los requisitos especificados en los apartados del análisis del sistema con los subsistemas propuestos para nuestro diseño.

Además, al finalizar la exposición de todos los diagramas de clases del sistema, expondremos una matriz de trazabilidad entre los componentes del sistema y los requisitos software catalogados en las actividades anteriores propias del análisis del sistema, de manera que podamos comprobar la consecución de todas las especificaciones requeridas por el cliente y todas las funcionalidades especificadas para el sistema.

Dichas tablas, seguirán el siguiente formato:

Identificador: CD-S-XY	
Nombre	<i>Nombre del Componente del Sistema</i>
Descripción	<i>Breve Descripción del Propósito que cumple dentro de la aplicación</i>
Subsistemas Relacionados	<i>Componentes que se encuentran relacionados en dependencia con el componente que estamos describiendo.</i>
Requisitos Software Relacionados	<i>Requisitos del Sistema que se encuentran relacionados con el componente actual, de manera que éste los satisface directamente, o participa en la consecución de su satisfacción.</i>
Métodos Públicos	<i>Funciones que proporciona al resto de componentes del sistema para el correcto funcionamiento del mismo.</i>

Tabla 76: Especificación del formato de nuestro catálogo de componentes del Sistema

Donde,

- CD: indica que se trata de un Componente del Diseño
- S: indica el subsistema al que pertenece. Pudiendo ser:
 - o P para el Subsistema presentación
 - o O para el Subsistema de Operaciones (O-W, para los componentes /clases del subsistema de procesamiento en *background*)
 - o D para el Subsistema de Datos (D-EN para los componentes /clases del subsistema entidades, D-AD para los del subsistema de Acceso a Datos)
- XY: son dos cifras que representan un valor numérico con el que enumeraremos de forma incremental todos los componentes/clases del subsistema correspondiente contemplados en nuestro catálogo, empezando por el 01.

SUBSISTEMA DE DATOS

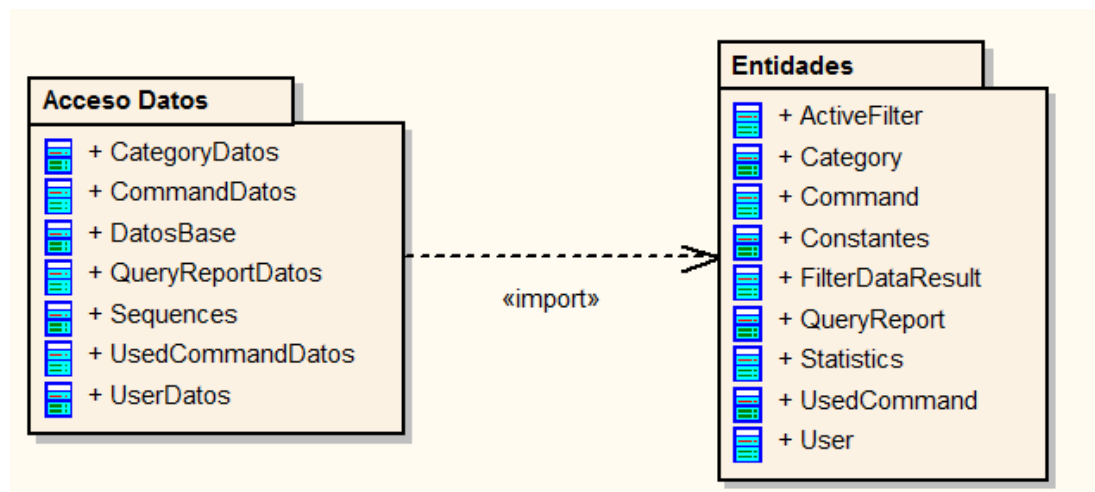


Ilustración 43: Diagrama Completo del Subsistema **Datos** - Componente Modelo

El Subsistema Datos en general, va a ser el módulo encargado de contener toda la lógica de almacenamiento de datos del sistema, la cual la repartirá directamente en dos subsistemas distintos: el de entidades, y el de acceso a datos.

Veamos cada uno de ellos.

SUBSISTEMA DE DATOS.ENTIDADES

El subsistema entidades, está constituido por todas aquellas clases que definen los *beans* del sistema, es decir, los objetos más básicos del mismo, las entidades del modelo entidad/relación anteriormente presentado; y, la clase constantes, que servirá básicamente para contener aquellos atributos constantes útiles para todo el sistema, abstrayendo su declaración y haciendo un uso común para el resto de componentes.

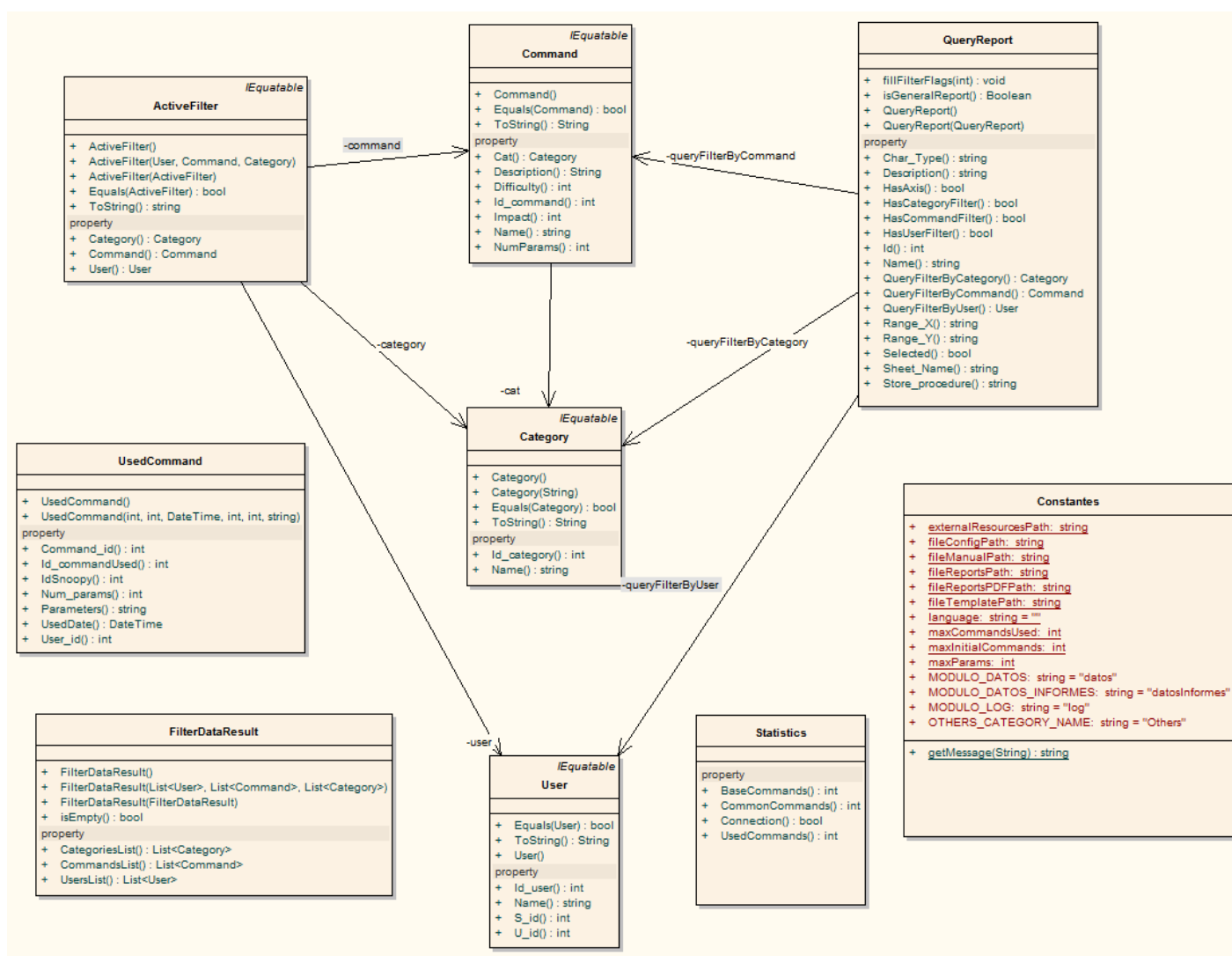


Ilustración 44: Diagrama Completo del Subsistema **Datos.Entidades** - Componente Modelo

Identificador: CD-D-EN-01	
Nombre	entidades.Command
Descripción	Clase que se utiliza para representar un comando Linux, con toda su información relevante para el análisis.
Subsistemas Relacionados	datos.CommandDatos, process.ChargeData, process.InitialCharge, process.FileTreatment
Requisitos Software	RS-F02
Métodos Públicos	Métodos para poder obtener y modificar los atributos de esta entidad y un método para comparar dos comandos.

Tabla 77: Descripción del Componente de Diseño D-EN-01 (Clase Command)

Identificador: CD-D-EN-02	
Nombre	entidades.Category
Descripción	Clase que se utiliza para representar una categoría de un comando, con toda su información relevante para el análisis.
Subsistemas Relacionados	datos.CategoryDatos, process.ChargeData, process.InitialCharge
Requisitos Software	RS-F02
Métodos Públicos	Métodos para poder obtener y modificar los atributos de esta entidad.

Tabla 78: Descripción del Componente de Diseño D-EN-02 (Clase Category)

Identificador: CD-D-EN-03	
Nombre	entidades.QueryReport
Descripción	Clase que se utiliza para representar los posibles informes que realizará el sistema, con su información relevante para el análisis.
Subsistemas Relacionados	datos.QueryReportDatos, forms.Form2, process.Report, process.workers.GenerateReportWorker
Requisitos Software	RS-F04, RS-F05, RS-F06, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RN02
Métodos Públicos	Métodos para poder obtener y modificar los atributos de esta entidad.

Tabla 79: Descripción del Componente de Diseño D-EN-03 (Clase QueryReport)

Identificador: CD-D-EN-04	
Nombre	entidades.UsedCommand
Descripción	Clase que se utiliza para representar los registros <i>log</i> y toda la información referente a los mismos, útil para el posterior análisis.
Subsistemas Relacionados	datos.UsedCommandDatos, process.ChargeData, process.FileTreatment
Requisitos Software	RS-F03
Métodos Públicos	Métodos para poder obtener y modificar los atributos de esta entidad.

Tabla 80: Descripción del Componente de Diseño D-EN-04 (Clase UsedCommand)

Identificador: CD-D-EN-05	
Nombre	entidades.User
Descripción	Clase que se utiliza para representar a los usuarios y su información relevante para el análisis.
Subsistemas Relacionados	datos.UserDatos, process.ChargeData, process.FileTreatment
Requisitos Software	RS-F03
Métodos Públicos	Métodos para poder obtener y modificar los atributos de esta entidad y un método para comparar dos usuarios.

Tabla 81: Descripción del Componente de Diseño D-EN-05 (Clase User)

Identificador: CD-D-EN-06	
Nombre	entidades.Constantes
Descripción	Clase que se utiliza para representar las constantes de la aplicación necesarias para su correcto funcionamiento.
Subsistemas Relacionados	datos.DatosBase, forms.Form1, forms.Form2, process.ChargeData, process.FileTreatment, process.InitialCharge, process.Report, process.worker.AbstractWorker, Program
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-F05, RS-F06, RS-NF-OP01, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04
Métodos Públicos	No tiene. Al ser constantes, se accede directamente a ellas, y en principio no deberían modificarse nunca.

Tabla 82: Descripción del Componente de Diseño D-EN-06 (Clase Constantes)

Identificador: CD-D-EN-07	
Nombre	entidades.Statistics
Descripción	Clase que se utiliza para saber si está disponible la conexión con el sistema gestor de bases de datos y para obtener una información general sobre el número de registros que hay en las tablas del sistema.
Subsistemas Relacionados	Forms.Form1, process.ChargeData, process.ProcessResult
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-NF-IN01, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Métodos para poder obtener y modificar los atributos.

Tabla 83: Descripción del Componente de Diseño D-EN-07 (Clase Statistics)

Identificador: CD-D-EN-08	
Nombre	entidades.FilterDataResult
Descripción	Clase que tiene tres listados con los usuarios, categorías y comandos, que podrán ser seleccionados por el usuario en función de los filtros seleccionados.
Subsistemas Relacionados	forms.Form2, process.workers.UpdateReportFiltersWorker
Requisitos Software	RS-F04, RS-NF-OP02, RS-NF-OP03
Métodos Públicos	Métodos generales de construcción, métodos para el acceso y modificación de los atributos de la clase, y un método para saber si está vacío o no.

Tabla 84: Descripción del Componente de Diseño D-EN-08 (Clase FilterDataResult)

Identificador: CD-D-EN-09	
Nombre	entidades.ActiveFilter
Descripción	Clase que contiene los datos de los filtros a aplicar en los informes.
Subsistemas Relacionados	forms.Form2, process.workers.UpdateReportFiltersWorker
Requisitos Software	RS-F04, RS-NF-OP02, RS-NF-OP03
Métodos Públicos	Métodos generales de construcción y actualización de los datos de los filtros, junto con un método para comparar dos clases de este tipo, y otro para obtener una cadena de texto con las descripciones de los filtros a aplicar.

Tabla 85: Descripción del Componente de Diseño D-EN-09 (Clase ActiveFilter)

SUBSISTEMA DE DATOS.ACCESO A DATOS

El subsistema de acceso a datos, como su propio nombre indica, contiene todas aquellas clases imprescindibles para que el sistema pueda interactuar con la base de datos del sistema, abstrayendo y encapsulando por tipo de entidad todos los métodos de inserción, consulta, borrado y/o modificación de los datos almacenados en las distintas tablas de la misma.

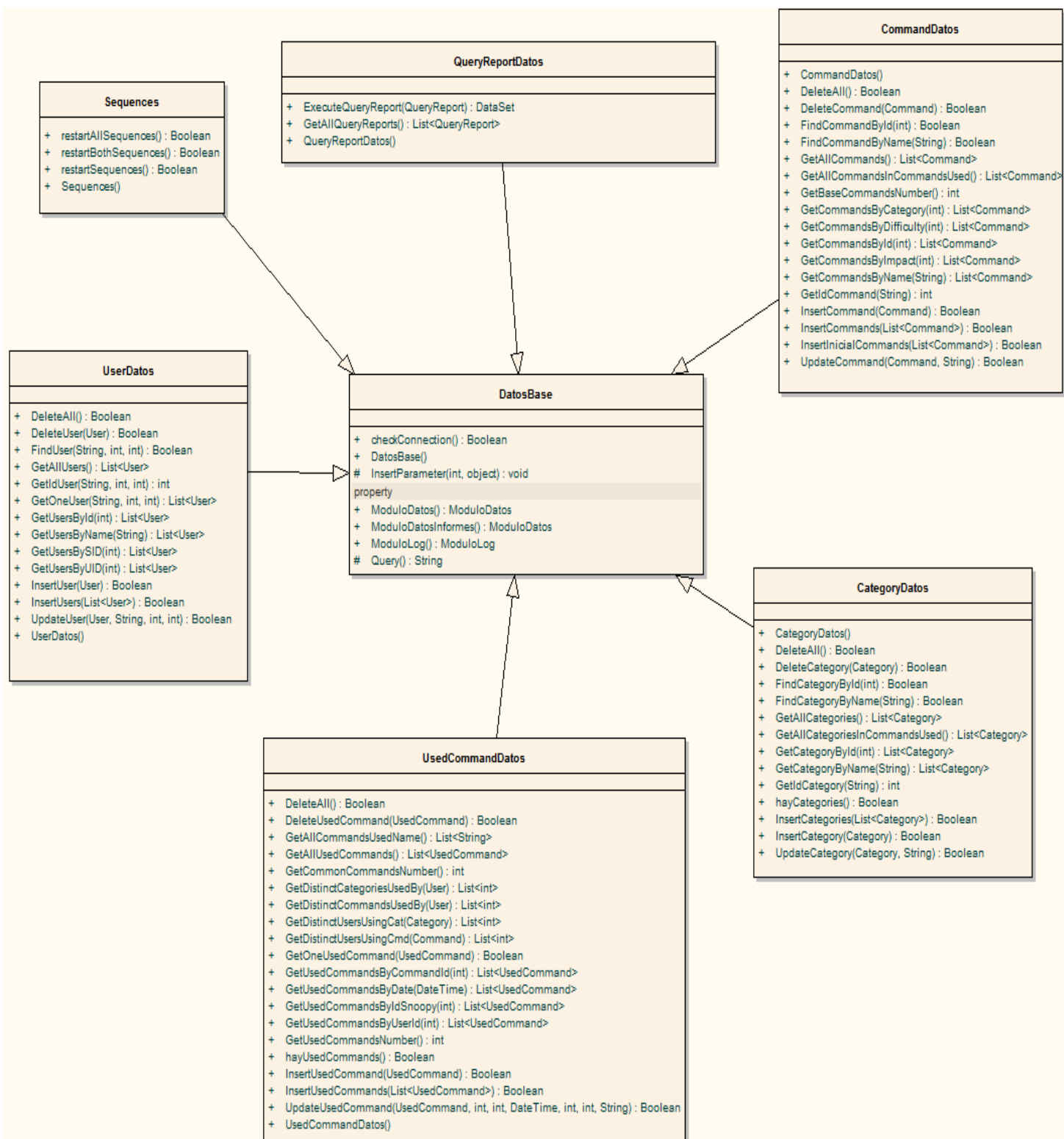


Ilustración 45: Diagrama Completo Subsistema *Datos.Acceso a Datos* - Componente Modelo

Identificador: CD-D-AD-01	
Nombre	datos.DatosBase
Descripción	Clase que sirve de base para todos los componentes del subsistema de acceso a datos.
Subsistemas Relacionados	datos.QueryReportDatos, datos.UsedCommandDatos, datos.CommandDatos, datos.CategoryDatos, datos.UserDatos, datos.Sequences
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-F05, RS-F06, RS-NF-OP01, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RN01, RS-NF-RN02
Métodos Públicos	Métodos para proporcionar a los componentes del subsistema datos aquellos atributos y funcionalidades comunes para todos.

Tabla 86: Descripción del Componente de Diseño D-AD-01 (Clase DatosBase)

Identificador: CD-D-AD-02	
Nombre	datos.CommandDatos
Descripción	Clase que representa los comandos de la base de datos del sistema. Contiene las funciones necesarias para trabajar con su tabla asociada.
Subsistemas Relacionados	process.ChargeData
Requisitos Software	RS-F01, RS-F02, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Métodos de consulta, inserción, borrado y actualización de los comandos almacenados en la base de datos.

Tabla 87: Descripción del Componente de Diseño D-AD-02 (Clase CommandDatos)

Identificador: CD-D-AD-03	
Nombre	datos.CategoryDatos
Descripción	Clase que representa las categorías de la base de datos del sistema. Contiene las funciones necesarias para trabajar con su tabla asociada.
Subsistemas Relacionados	process.ChargeData
Requisitos Software	RS-F01, RS-F02, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Métodos de consulta, inserción, borrado y actualización de las categorías almacenadas en la base de datos.

Tabla 88: Descripción del Componente de Diseño D-AD-03 (Clase CategoryDatos)

Identificador: CD-D-AD-04	
Nombre	datos.QueryReportDatos
Descripción	Clase que representa los tipos de informe almacenados en la base de datos del sistema. Contiene todas las funciones necesarias para trabajar con su tabla asociada.
Subsistemas Relacionados	forms.Form2, process.Report
Requisitos Software	RS-F04, RS-F05, RS-F06, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RN02
Métodos Públicos	Métodos de consulta, inserción, borrado y actualización de los tipos de informes almacenados en la base de datos, y métodos de ejecución de sus procedimientos almacenados.

Tabla 89: Descripción del Componente de Diseño D-AD-04 (Clase QueryReportDatos)

Identificador: CD-D-AD-05	
Nombre	datos.UsedCommandDatos
Descripción	Clase que sirve para representar los registros <i>log</i> almacenados en la base de datos del sistema. Contiene las funciones necesarias para trabajar con su tabla asociada.
Subsistemas Relacionados	process.ChargeData
Requisitos Software	RS-F01, RS-F03, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Métodos de consulta, inserción, borrado y actualización de los registros <i>log</i> almacenados en la base de datos.

Tabla 90: Descripción del Componente de Diseño D-AD-05 (Clase UsedCommandDatos)

Identificador: CD-D-AD-06	
Nombre	datos.UserDatos
Descripción	Clase que representa los usuarios de la base de datos del sistema; contiene las funciones necesarias para trabajar con su tabla asociada.
Subsistemas Relacionados	process.ChargeData
Requisitos Software	RS-F01, RS-F03, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Métodos de consulta, inserción, borrado y actualización de los usuarios almacenados en la base de datos.

Tabla 91: Descripción del Componente de Diseño D-AD-06 (Clase UserDatos)

Identificador: CD-D-AD-07	
Nombre	datos.Sequences
Descripción	Clase que contiene las funciones necesarias para trabajar con las secuencias de las tablas de la base de datos del sistema.
Subsistemas Relacionados	process.ChargeData
Requisitos Software	RS-F02, RS-F03
Métodos Públicos	Métodos para reiniciar las secuencias de las tablas de la base de datos.

Tabla 92: Descripción del Componente de Diseño D-AD-07 (Clase Sequences)

SUBSISTEMA DE PRESENTACIÓN

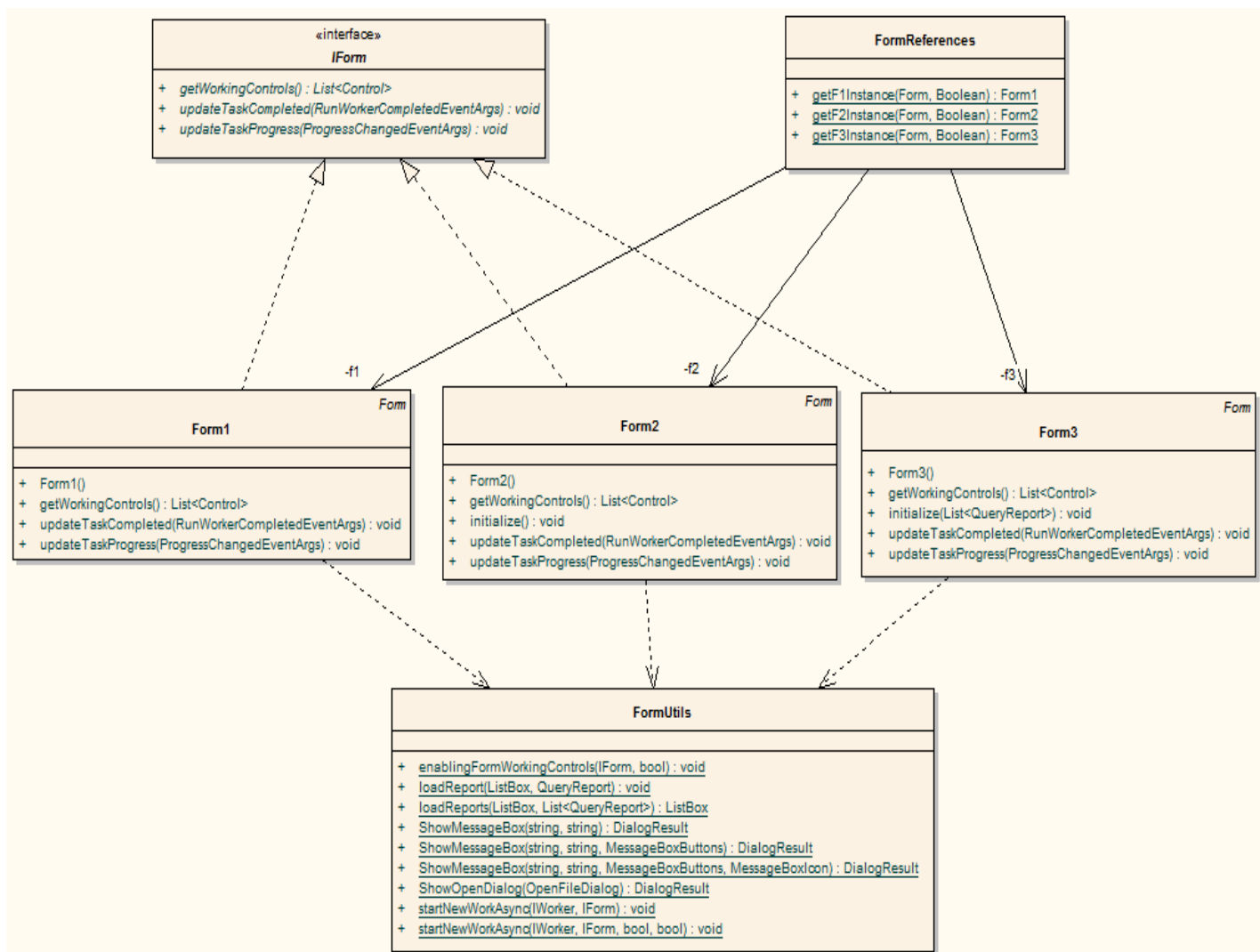


Ilustración 46: Diagrama de Clases Completo Subsistema **Presentación** - Componente Vista

El Subsistema de Presentación es el encargado de contener toda la lógica necesaria para permitir la interacción del usuario con el sistema.

En total consta de seis componentes bien diferenciados: tres principales (Form1, Form2, Form3), dos auxiliares (FormUtils, FormReferences), y una interfaz (IForm), encargado cada uno de ellos de una funcionalidad muy distinta pero complementaria a la de los demás, de manera que entre todos consiguen dotar a la aplicación de aquellas funcionalidades requeridas por el cliente y recogidas en los requisitos del catálogo correspondiente.

La navegabilidad entre los tres módulos principales es completa, permitiendo al usuario moverse entre los mismos de las siguientes maneras según el siguiente esquema:

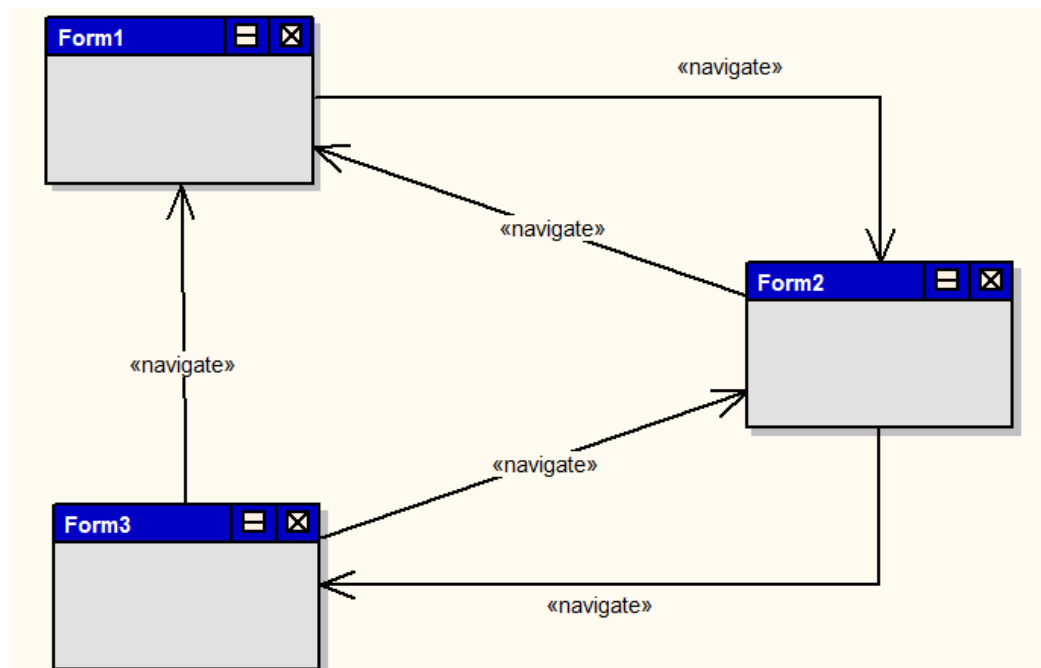


Ilustración 47: Esquema de Navegación entre Formularios

Siendo el Form1 el formulario de entrada y salida de la aplicación.

Veamos a continuación cada uno de ellos.

Identificador: CD-P-01	
Nombre	forms.Form1
Descripción	Formulario Principal que permite la interacción del usuario con el sistema.
Subsistemas Relacionados	forms.Form2, forms.Form3, Program
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-F05, RS-F06, RS-NF-IN01, RS-NF-IN02, RS-NF-IN03, RS-NF-IN04, RS-NF-IN05, RS-NF-OP01, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RS06
Métodos Públicos	Métodos que permiten la interacción propia del usuario con la aplicación.

Tabla 93: Descripción del Componente de Diseño P-01 (Clase Form1)

Identificador: CD-P-02	
Nombre	forms.Form2
Descripción	Es el Formulario de Generación de Informes y como su propio nombre indica permite la interacción del usuario con el sistema para la generación de los informes que desee, dándole las opciones de filtrado de datos y selección de informes.
Subsistemas Relacionados	forms.Form1, forms.Form3
Requisitos Software	RS-F04, RS-F05, RS-F06, RS-NF-IN01, RS-NF-IN02, RS-NF-IN03, RS-NF-IN04, RS-NF-IN05, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RS05, RS-NF-VA01
Métodos Públicos	Métodos que permiten la interacción propia del usuario con la aplicación y la inicialización de los elementos del formulario.

Tabla 94: Descripción del Componente de Diseño P-02 (Clase Form2)

Identificador: CD-P-03	
Nombre	forms.Form3
Descripción	Formulario de Tarea Finalizada que anuncia al usuario el fin de la tarea finalizada y le permite continuar interactuando con la aplicación.
Subsistemas Relacionados	forms.Form2
Requisitos Software	RS-F05, RS-NF-IN01, RS-NF-IN02, RS-NF-IN03, RS-NF-IN04, RS-NF-IN05
Métodos Públicos	Métodos que permiten la interacción propia del usuario con la aplicación y la inicialización de los elementos del formulario.

Tabla 95: Descripción del Componente de Diseño P-03 (Clase Form3)

Identificador: CD-P-04	
Nombre	Forms.IForm
Descripción	Clase que define los métodos que necesitarán los formularios para arrancar las tareas derivadas de la interacción del usuario con la aplicación en segundo plano.
Subsistemas Relacionados	forms.Form1, forms.Form2, forms.Form3
Requisitos Software	
Métodos Públicos	Métodos que tendrán que implementar los formularios de la aplicación para llevar a cabo las tareas solicitadas por el usuario.

Tabla 96: Descripción del Componente de Diseño P-04 (Clase IForm)

Identificador: CD-P-05	
Nombre	forms.FormReferences
Descripción	Clase encargada de crear los nuevos formularios solicitados, o devolver una instancia al formulario deseado en el caso de que este ya existiese y no se precisara la creación de uno nuevo. También se encarga de ocultar el formulario solicitante y mostrar el solicitado.
Subsistemas Relacionados	forms.Form1, forms.Form2, forms.Form3
Requisitos Software	RS-NF-IN01, RS-NF-IN04
Métodos Públicos	Métodos necesarios para poder solicitar/crear un formulario de los tres tipos principales de la aplicación.

Tabla 97: Descripción del Componente de Diseño P-05 (Clase FormReferences)

Identificador: CD-P-06	
Nombre	forms.FormUtilis
Descripción	Método de utilidades comunes a todos los formularios de la aplicación.
Subsistemas Relacionados	forms.Form1, forms.Form1, forms.Form3
Requisitos Software	RS-NF-IN01, RS-NF-IN02, RS-NF-IN03
Métodos Públicos	Métodos que permiten la interacción propia del usuario con la aplicación y la inicialización de los elementos del formulario.

Tabla 98: Descripción del Componente de Diseño P-06 (Clase FormUtilis)

SUBSISTEMA DE OPERACIONES

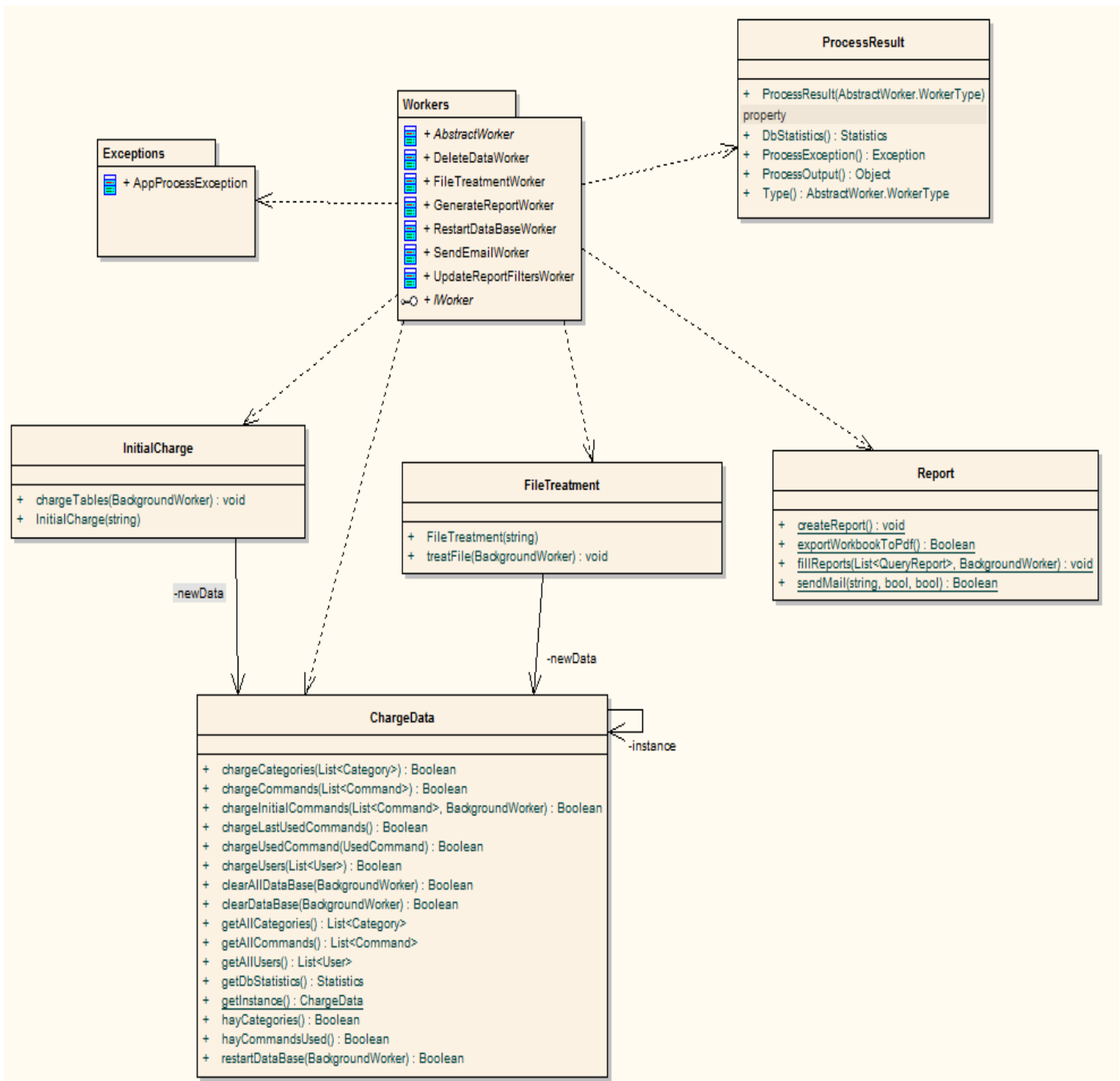


Ilustración 48: Diagrama de Clases del Subsistema **Operaciones** - Componente Controlador

El subsistema de operaciones, se podría decir que es el más importante de todo el sistema, ya que aparte de contener toda la lógica operacional del mismo, también es el encargado de capturar las peticiones dadas por el usuario a través del subsistema de presentación, y mandar las órdenes correspondientes para su satisfacción al subsistema de datos, enlazando en definitiva las necesidades e intenciones del cliente, con la base de datos del sistema, y por tanto con la información contenida en el mismo.

Identificador: CD-O-01	
Nombre	process.InitialCharge
Descripción	Clase que se utiliza para cargar en la base de datos el fichero de comandos iniciales proporcionado por el usuario.
Subsistemas Relacionados	process.InitialCharge, process.workers.RestartDataBaseWorker
Requisitos Software	RS-F02, RS-NF-RN01, RS-NF-RS01
Métodos Públicos	Proporciona todos los métodos necesarios para leer el fichero de comandos iniciales y cargar los comandos y las categorías en la base de datos del sistema.

Tabla 99: Descripción del Componente de Diseño O-01 (Clase InitialCharge)

Identificador: CD-O-02	
Nombre	process.FileTreatment
Descripción	Clase que se utiliza para cargar en la base de datos el fichero de <i>logs</i> que quiere analizar el usuario.
Subsistemas Relacionados	process.FileTreatment, process.workers.FileTreatmentWorker
Requisitos Software	RS-F03, RS-NF-RN01, RS-NF-RS02
Métodos Públicos	Proporciona todos los métodos necesarios para leer el fichero de <i>logs</i> y cargar los comandos usados y sus usuarios en la base de datos del sistema.

Tabla 100: Descripción del Componente de Diseño O-02 (Clase FileTreatment)

Identificador: CD-O-03	
Nombre	process.Report
Descripción	Clase que se utiliza para generar los informes solicitados por el usuario.
Subsistemas Relacionados	forms.Form2, process.Report, process.GenerateReportWorker
Requisitos Software	RS-F04, RS-F05, RS-F06, RS-NF-OP02, RS-NF-OP03, RS-NF-OP04, RS-NF-RN02, RS-NF-RS03, RS-NF-RS05, RS-NF-VA01
Métodos Públicos	Proporciona todos los métodos necesarios para generar los informes solicitados por el usuario, exportarlos a PDF, mostrárselos por pantalla y/o enviarlos por e-mail.

Tabla 101: Descripción del Componente de Diseño O-03 (Clase Report)

Identificador: CD-O-04	
Nombre	process.ChargeData
Descripción	Clase auxiliar que sirve principalmente para relacionar al subsistema de operaciones con el subsistema de datos.
Subsistemas Relacionados	forms.Form1, process.FileTreatment, process.InitialCharge, process.workers.AbstractWorker
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-NF-OP01, RS-NF-RN01
Métodos Públicos	Proporciona todos los métodos necesarios para solicitar la consulta, inserción, borrado y actualización de los distintos registros almacenados en la base de datos: comandos, categorías, usuarios y comandos usados o <i>logs</i> , y el reinicio de las distintas secuencias de las tablas.

Tabla 102: Descripción del Componente de Diseño O-04 (Clase ChargeData)

Identificador: CD-O-05	
Nombre	process.ProcessResult
Descripción	Clase que contiene los datos que se volverán a los formularios como resultado de una tarea ejecutada en segundo plano.
Subsistemas Relacionados	process.workers
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-NF-OP01, RS-NF-OP02
Métodos Públicos	Métodos propios de creación de la clase, y métodos para proporcionar el acceso y modificación de los atributos de la misma.

Tabla 103: Descripción del Componente de Diseño O-05 (Clase ProcessResult)

Identificador: CD-O-06	
Nombre	process.Exceptions.AppProcessException
Descripción	Clase que se encarga de capturar y dar formato a las excepciones propias de la ejecución en segundo plano de los <i>workers</i> , de manera que se le pueda informar con detalle de las mismas a los formularios de la aplicación y por ende al usuario.
Subsistemas Relacionados	process.workers
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-NF-OP01, RS-NF-OP02, RS-NF-OP04
Métodos Públicos	Métodos propios de creación y métodos propios de acceso a los atributos de la clase.

Tabla 104: Descripción del Componente de Diseño O-06 (Clase AppProcessException)

SUBSISTEMA DE OPERACIONES.PROCESAMIENTO EN BACKGROUND (WORKERS)

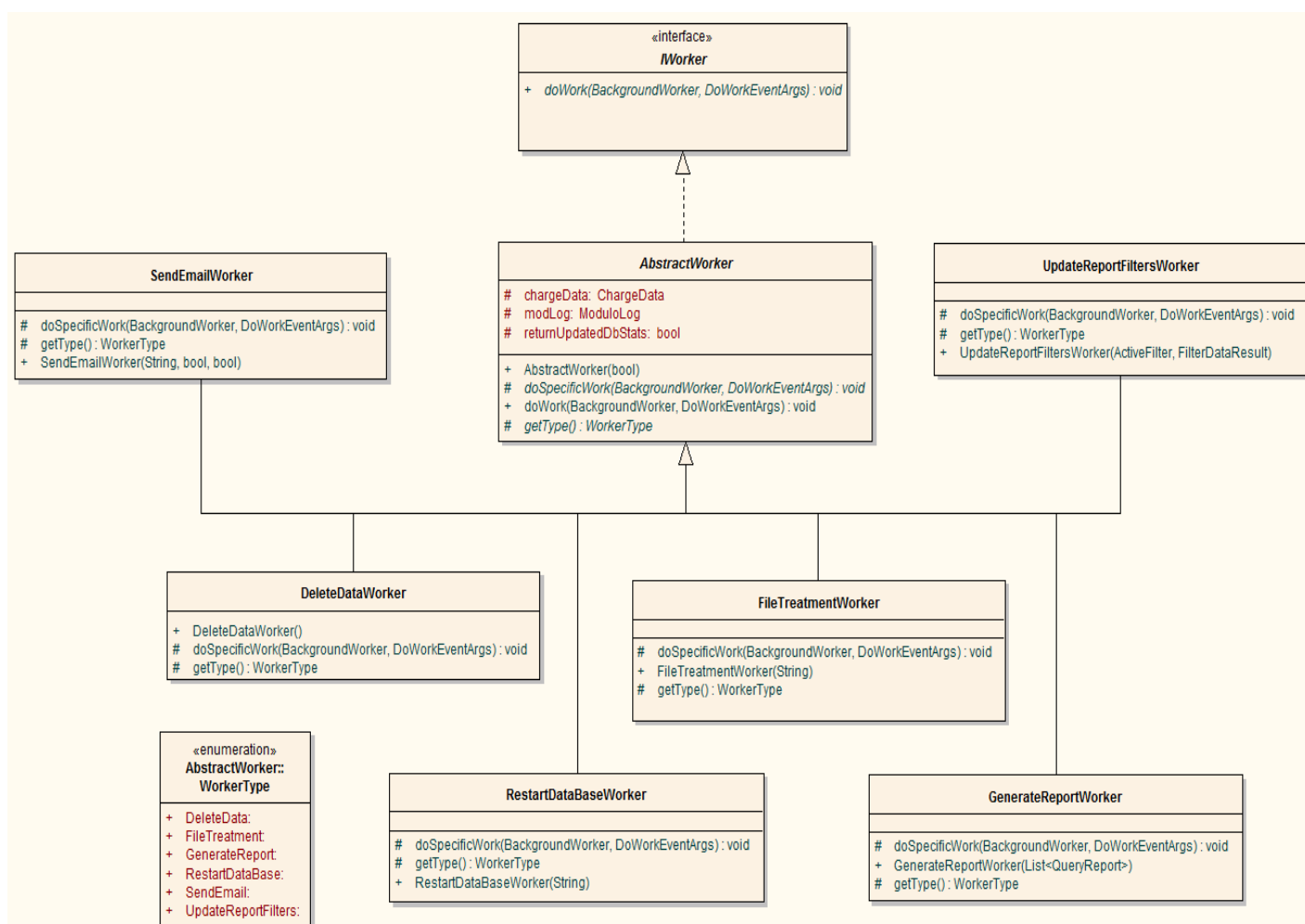


Ilustración 49: Diagrama Subsistema *Procesamiento Background* - Componente Controlador

El subsistema de procesamiento en *background*, o subsistema “*workers*”, constituye la encapsulación de todas aquellas funcionalidades que suponen la ejecución de tareas que llevan un mayor consumo de tiempo y recursos, facilitando así su ejecución paralela en segundo plano de forma asíncrona con la interfaz y, evitando su bloqueo. Además, nos permiten ir informando al usuario del progreso de la tarea en ejecución.

Identificador: CD-O-W-01	
Nombre	process.workers.IWorker
Descripción	Interfaz para la creación de los distintos <i>Workers</i> del sistema.
Subsistemas Relacionados	forms.Form1, forms.Form2, process.workers.AbstractWorker
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-F05, RS-F06, RS-NF-OP01, RS-NF-OP04, RS-NF-RS04, RS-NF-RS05
Métodos Públicos	Definición del método doWork() para todos los workers.

Tabla 105: Descripción del Componente de Diseño O-W-01 (Clase IWorker)

Identificador: CD-O-W-02	
Nombre	process.workers.AbstractWorker
Descripción	Clase abstracta con la funcionalidad común para los distintos workers del sistema.
Subsistemas Relacionados	process.workers.DeleteDataWorker, process.workers.FileTreatmentWorker, process.workers.GenerateReportWorker, process.workers.RestartDataBaseWorker
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-F05, RS-F06, RS-NF-OP01, RS-NF-OP04, RS-NF-RS04, RS-NF-RS05
Métodos Públicos	Proporciona el constructor común a todos los workers con sus atributos respectivos.

Tabla 106: Descripción del Componente de Diseño O-W-02 (Clase AbstractWorker)

Identificador: CD-O-W-03	
Nombre	process.workers.DeleteDataWorker
Descripción	Clase que ejecuta en background el borrado de toda la información almacenada en la base de datos.
Subsistemas Relacionados	forms.Form1
Requisitos Software	RS-F01, RS-NF-OP01
Métodos Públicos	Implementa el método doWork para ejecutar el borrado de la base de datos en segundo plano.

Tabla 107: Descripción del Componente de Diseño O-W-03 (Clase DeleteDataWorker)

Identificador: CD-O-W-04	
Nombre	process.workers.GenerateReportWorker
Descripción	Clase que ejecuta en background la generación del informe solicitado por el usuario.
Subsistemas Relacionados	forms.Form2
Requisitos Software	RS-F04, RS-F05, RS-F06, RS-NF-OP04, RS-NF-RS05
Métodos Públicos	Implementa el método doWork para ejecutar la generación del informe solicitado en segundo plano.

Tabla 108: Descripción del Componente de Diseño O-W-04 (Clase GenerateReportWorker)

Identificador: CD-O-W-05	
Nombre	process.workers.FileTreatmentWorker
Descripción	Clase que ejecuta en background la carga del fichero de <i>logs</i> proporcionado por el usuario.
Subsistemas Relacionados	forms.Form1
Requisitos Software	RS-F03, RS-NF-RS04
Métodos Públicos	Implementa el método doWork para ejecutar la carga del fichero de <i>logs</i> proporcionado por el usuario en la base de datos del sistema, en segundo plano.

Tabla 109: Descripción del Componente de Diseño O-W-05 (Clase FileTreatmentWorker)

Identificador: CD-O-W-06	
Nombre	process.workers.RestartDataBaseWorker
Descripción	Clase que ejecuta en background la carga del fichero de comandos iniciales proporcionado por el usuario.
Subsistemas Relacionados	forms.Form1
Requisitos Software	RS-F02
Métodos Públicos	Implementa el método doWork para ejecutar la carga del fichero de comandos iniciales proporcionado por el usuario en la base de datos del sistema, en segundo plano.

Tabla 110: Descripción del Componente de Diseño O-W-06 (Clase RestartDataBaseWorker)

Identificador: CD-O-W-07	
Nombre	process.workers.SendEmailWorker
Descripción	Clase que ejecuta en <i>background</i> el envío de los informes a la dirección de correo electrónico proporcionada por el usuario.
Subsistemas Relacionados	forms.Form3
Requisitos Software	RS-F05, RS-NF-OP04
Métodos Públicos	Implementa el método doWork para ejecutar el envío de los informes por e-mail, en segundo plano.

Tabla 111: Descripción del Componente de Diseño O-W-07 (Clase SendEmailWorker)

Identificador: CD-O-W-08	
Nombre	process.workers.UpdateReportsFilterWorker
Descripción	Clase que ejecuta en <i>background</i> la actualización de los datos mostrados en los <i>combos</i> de los formularios en función de los filtros marcados.
Subsistemas Relacionados	forms.Form2
Requisitos Software	RS-F04, RS-NF-OP02, RS-NF-OP03
Métodos Públicos	Implementa el método <i>doWork</i> para ejecutar la actualización de los <i>combos</i> del formulario de forma imperceptible por el usuario.

Tabla 112: Descripción de Componente de Diseño O-W-08 (Clase UpdateReportsFilterWorker)

Identificador: CD-O-W-09	
Nombre	process.workers.WorkerType
Descripción	Enumerado para identificar el tipo de <i>worker</i> que genera la excepción.
Subsistemas Relacionados	process.workers, forms.Form2
Requisitos Software	RS-F01, RS-F02, RS-F03, RS-F04, RS-NF-OP01, RS-NF-OP02, RS-NF-OP04
Métodos Públicos	Al ser un enumerado, no tienen ningún método público implementado.

Tabla 113: Descripción del Componente de Diseño O-W-09 (Clase RestartDataBaseWorker)

MATRICES DE TRAZABILIDAD: Componentes – Requisitos Software

Una vez vistos y analizados todos los componentes que formarán parte de nuestro sistema, vamos a exponer su matriz de trazabilidad con los requisitos software especificados en el apartado de Análisis del Sistema de Información, de manera que podamos dar por completado el diseño, y podamos pasar a transformar estos componentes en su correspondiente proyecto C# [17].

	Acceso Datos::CategoryDatos	Acceso Datos::CommandDatos	Acceso Datos::DatosBase	Acceso Datos::QueryReportDatos	Acceso Datos::Sequences	Acceso Datos::UsedCommandDatos	Acceso Datos::UserDatos	Class Model::BackgroundWorker	Entidades::ActiveFilter	Entidades::Category	Entidades::Command	Entidades::Constantes	Entidades::FilterDataResult	Entidades::QueryReport	Entidades::Statistics	Entidades::UsedCommand	Entidades::User	Exceptions::AppProcessException	Forms::Form1	Forms::Form2	Forms::Form3	Forms::FormReferences	Forms::FormUtils	Forms::Messages	Process::ChargeData	Process::FileTreatment	Process::InitialCharge	Process::ProcessResult	Process::Report	Workers::AbstractWorker	Workers::DeleteDataWorker	Workers::FileTreatmentWorker	Workers::GenerateReportWorker	Workers::RestartDataBaseWorker	Workers::SendEmailWorker	Workers::UpdateReportFiltersWorker	Workers::WorkerType
Functional Requirements::RS-F01:...	X	X	X		X	X	X					X			X			X	X						X			X		X	X						X
Functional Requirements::RS-F02:...	X	X	X		X			X		X	X	X			X			X	X						X		X	X		X				X			X
Functional Requirements::RS-F03:...			X			X	X	X					X		X			X	X						X	X		X		X		X					X
Functional Requirements::RS-F04:...			X	X				X	X	X	X	X	X	X	X		X	X	X	X	X				X			X	X	X			X			X	X
Functional Requirements::RS-F05:...			X	X						X	X	X		X			X		X	X	X								X	X			X		X		
Functional Requirements::RS-F06:...			X	X								X		X					X	X									X	X			X				

Tabla 114: Trazabilidad entre los Componentes y los Requisitos Funcionales del Sistema

	Acceso Datos::CategoryDatos	Acceso Datos::CommandDatos	Acceso Datos::DatosBase	Acceso Datos::QueryReportDatos	Acceso Datos::Sequences	Acceso Datos::UsedCommandDatos	Acceso Datos::UserDatos	Class Model::BackgroundWorker	Entidades::ActiveFilter	Entidades::Category	Entidades::Command	Entidades::Constantes	Entidades::FilterDataResult	Entidades::QueryReport	Entidades::Statistics	Entidades::UsedCommand	Entidades::User	Exceptions::AppProcessException	Forms::Form1	Forms::Form2	Forms::Form3	Forms::FormReferences	Forms::FormUtils	Forms::Messages	Process::ChargeData	Process::FileTreatment	Process::InitialCharge	Process::ProcessResult	Process::Report	Workers::AbstractWorker	Workers::DeleteDataWorker	Workers::FileTreatmentWorker	Workers::GenerateReportWorker	Workers::RestartDataBaseWorker	Workers::SendEmailWorker	Workers::UpdateReportFiltersWorker	Workers::WorkerType
Non-Functional Requirements::RS-NF-DI01: ...																																					
Non-Functional Requirements::RS-NF-DI02: ...																																					
Non-Functional Requirements::RS-NF-DI03: ...																																					
Non-Functional Requirements::RS-NF-IN01:...								X							X					X	X	X	X	X	X												
Non-Functional Requirements::RS-NF-IN02:...												X								X	X	X		X	X												
Non-Functional Requirements::RS-NF-IN03:...																				X				X													
Non-Functional Requirements::RS-NF-IN04:...																				X	X	X	X														
Non-Functional Requirements::RS-NF-IN05:...																						X															
Non-Functional Requirements::RS-NF-OP01:...	X	X	X		X	X	X	X		X	X	X			X		X	X	X	X					X			X		X	X			X		X	
Non-Functional Requirements::RS-NF-OP02:...			X	X				X	X	X	X	X	X	X		X	X	X	X	X					X			X	X						X	X	
Non-Functional Requirements::RS-NF-OP03:...			X	X				X	X	X	X	X	X	X						X	X							X					X			X	
Non-Functional Requirements::RS-NF-OP04:...			X	X				X				X		X				X	X	X								X	X					X		X	
Non-Functional Requirements::RS-NF-RN01:...	X	X	X			X	X								X										X	X	X										
Non-Functional Requirements::RS-NF-RN02:...			X	X									X																X								
Non-Functional Requirements::RS-NF-RS01:...																											X										
Non-Functional Requirements::RS-NF-RS02:...																										X											
Non-Functional Requirements::RS-NF-RS03:...																				X								X					X				
Non-Functional Requirements::RS-NF-RS04:...																				X						X				X		X					
Non-Functional Requirements::RS-NF-RS05:...																					X							X	X				X				
Non-Functional Requirements::RS-NF-RS06:...																				X																	
Non-Functional Requirements::RS-NF-VA01:...																					X								X					X			

Tabla 115: Trazabilidad entre los Componentes y los Requisitos No Funcionales del Sistema

Como se puede apreciar, hay 3 requisitos del sistema que no tienen trazabilidad directa hasta los componentes del diseño que hemos extraído. Son los denominados requisitos de disponibilidad.

Esto, en un principio puede parecer una inconsistencia en el diseño; pero, si lo analizamos detalladamente, podemos ver cómo se trata principalmente de unos requisitos que no han de ser necesariamente cubiertos por ningún componente específico de los que se han descrito en los puntos anteriores, sino que simplemente deberán cumplirse a la hora de implementar e implantar el sistema actual, consiguiendo su satisfacción y realización.

Son requisitos que competen al funcionamiento del sistema en conjunto, es decir, como un todo, por lo que es normal que no se corresponda con ningún componente específico de forma individualizada, ni con ninguna funcionalidad concreta.

5.4.3 Diseño de la Jerarquía

(Métrica v.3 [12] Tarea DSI 4.5)

Esta tarea tiene como objetivo exponer y revisar la jerarquía de clases resultante del modelo de clases obtenido en las tareas anteriores y comprobar que es viable en el entorno de desarrollo que vamos a utilizar.

En C# [17] una clase puede implementar varias interfaces de forma simultánea, sin embargo la herencia múltiple no está permitida, por lo que las relaciones de jerarquía presentes en el diseño se tendrán que adaptar a estas restricciones.

Jerarquías presentes en el modelo:

- Entidades que implementan la interfaz **IEquatable**.
- Entidad **DatosBase** de la que extienden los módulos de acceso a la base de datos.
- Formularios que extienden de la clase **Form**.
- Interfaz y clase abstracta en el subsistema “workers”.

En primer lugar, y por estar desarrollando una aplicación cuya interfaz de usuario está realizada mediante Windows Forms, la primera jerarquía que podremos observar es la que existe entre todos los **formularios** de la aplicación y la clase **Form**, que contiene los métodos y atributos necesarios para la correcta creación y ejecución de los mismos y sus elementos.

Otra jerarquía importante es la existente en el subsistema “**workers**”, donde existirá una **interfaz** con el contrato expuesto por los elementos de este subsistema al exterior, y una **clase abstracta** con la implementación común a todos los elementos. Luego, la jerarquía será simplemente la extensión de dicha clase abstracta por cada uno de los “workers” contenidos en la aplicación.

En el subsistema de **datos** también existirá otra jerarquía, ya que todas las clases que proporcionan a la aplicación el acceso a la capa de datos **extenderán** de la clase **DatosBase**, que será la que proporcionará la implementación común, y la gestión de la conexión con la base de datos.

Por último y aunque sea una jerarquía bastante simple, algunos modelos del subsistema de **entidades** implementarán la interfaz **IEquatable**, de forma que sea posible comparar los datos de entidades del mismo tipo de forma sencilla, y a la hora de importar los datos del fichero de *log* de origen podamos utilizar estas simples comprobaciones de forma directa a través de los propios modelos.

Conclusión: Las jerarquías propuestas en el diseño de clases presentado, es viable en el entorno de desarrollo que vamos a utilizar y por ende, nos servirá perfectamente como base para la implementación del sistema que queremos implementar.

5.4.4 Especificación de Necesidad de Migración y Carga Inicial de Datos

(Métrica v.3 [12] Tarea DSI 4.7)

Tal y como se especificó en el apartado 4.3.2 del presente documento (Especificación de Necesidades de Migración de Datos y Carga Inicial) para el correcto funcionamiento del sistema de información que estamos diseñando, no es necesaria ninguna migración de datos, puesto que no existe ningún sistema previo del que migrar información; pero sí que existe la necesidad de hacer una carga inicial de datos, por lo que el objetivo principal de esta tarea será realizar una especificación general de esta necesidad.

La carga inicial de datos que se requiere para el correcto funcionamiento del sistema es básicamente la creación de las estructuras de almacenamiento de los datos dentro de nuestra base de datos (tablas, secuencias, procedimientos almacenados, etc.), y la carga de la información necesaria referente a los tipos de informes que se van a poder realizar con el sistema de información que estamos diseñando (tabla QueryReports).

Para la realización de esta carga, se proporcionará un script SQL que realizará las tareas necesarias de creación, inserción y actualización, en la base de datos.

Dicha carga consistirá básicamente en realizar los siguientes pasos:

1. Creación de las tablas necesarias para el almacenamiento de los datos de nuestro sistema.
2. Creación de las secuencias de cada una de estas tablas.
3. Creación de los procedimientos almacenados propios de nuestro sistema.
4. Creación de la tabla QUERY_REPORTS
5. Inserción de los tipos de informes que queremos realizar.
6. Creación de los procedimientos almacenados propios de cada tipo de informe.

La tabla Query_Reports, es una estructura esencial para nuestro sistema, ya que es quien le indica a la aplicación qué tipos de informe debe realizar, y cómo.

Veámosla un poco más en detalle. La tabla Query Report tendrá los siguientes campos:

ID_Informe	<i>Número entero identificador del informe al que se refiere</i>
Nombre	<i>Nombre específico del Informe</i>
Descripción	<i>Descripción de lo que va a analizar el informe</i>
PL/SQL	<i>Procedimiento almacenado con la lógica del informe</i>
Tipo de Gráfico	<i>Tipo de gráfico Excel del informe a generar</i>
Plantilla	<i>Plantilla que se va a utilizar para generar el informe</i>
Filtros	<i>Tipo de filtro que admite el informe: por usuario y/o por comando y/o por categoría.</i>

Tabla 116: Definición de la Tabla Query Reports

Esta tabla, vendrá siempre cargada desde el principio en la base de datos del sistema, ya que su acceso a la misma desde la aplicación, es únicamente de lectura, estando la funcionalidad principal de la misma supeditada a la existencia y correcta ejecución de los datos almacenados en ella.

La lógica de análisis del sistema, está directamente relacionada con los registros almacenados en esta tabla, y más concretamente, con la correcta ejecución de los PL's referentes a cada uno de ellos, por lo que la existencia de la tabla correctamente cumplimentada en el sistema, antes de la ejecución de la aplicación, es de imprescindible cumplimiento para el correcto funcionamiento de la misma.

La tabla de informes, a su vez se apoya de dos tablas secundarias que le proporcionan datos complementarios al tipo de gráfico que va a tener el informe y a la plantilla que se va a utilizar para generarlo.

5.5 Diseño Físico de Datos

Este apartado se corresponde con la Actividad DSI 6 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta actividad tiene como objetivo principal definir la estructura física de los datos que va a utilizar el sistema de información que estamos diseñando. Dicha estructura se realizará tomando como punto de partida el modelo de clases obtenido en las actividades anteriores, los requisitos establecidos para el sistema, las características específicas del mismo, y las especificaciones del entorno tecnológico que lo soporta, consiguiendo así una mayor eficiencia en el tratamiento de los datos, mejorando los tiempos de respuesta y optimizando los recursos de la máquina.

5.5.1 Diseño del Modelo Físico de Datos

(Métrica v.3 [12] Tarea DSI 6.1)

El objetivo de esta tarea es realizar el modelo físico de datos del sistema que estamos diseñando a partir del modelo de clases realizado en apartados anteriores y teniendo en cuenta las peculiaridades técnicas del gestor de bases de datos elegido para el desarrollo del mismo.

- Características y Particularidades del Sistema Gestor de Bases de Datos elegido

Oracle 10g [23] es un sistema gestor de base de datos destacado principalmente por su alto rendimiento, incluyendo escalabilidad, robustez, confiabilidad, disponibilidad, un desarrollo de fácil despliegue, una plataforma Oracle [23] que incluye SQL, PL/SQL, Java y XML e interconectividad, entre otros.

Según su creador (Oracle [23]) este sistema gestor proporciona los siguientes BENEFICIOS:

1. Para organizaciones soportando entre 1-1000 usuarios.
2. Bases de datos hasta aproximadamente 500 GB de datos.
3. Soporte para sistemas altamente disponibles.
4. Simple configuración e instalación, administración automatizada.
5. Compatible con todo tipo de datos y con todas las aplicaciones.
6. Desempeño, disponibilidad, seguridad y confiabilidad comprobada.

Además, algunas de sus características principales que hay que tener en cuenta porque atañen directamente a la creación de cualquier modelo físico de datos son:

- ✓ Permite la definición y el uso de Disparadores
 - ✓ Permite la especificación de *Checks* para las restricciones por fila
 - ✓ Permite el borrado en cascada
 - ✓ Permite el borrado restringido
 - ✓ Permite la puesta a NULL de los valores
 - ✓ Permite la modificación restringida
 - ✓ Permite la definición y uso de Secuencias
 - ✓ Admite la definición y uso de procedimientos almacenados (PL's)
 - ✗ No permite la definición y uso de aserciones como tal, aunque sí que se pueden implementar con el uso de disparadores
 - ✗ No permite la modificación en cascada
- Estimación sobre el uso y volumen de los datos de cada entidad / clase

Dentro de nuestro sistema, hay dos entidades principales que, en función del uso que se le den, pueden llegar a almacenar una gran cantidad de datos. Estas entidades son:

- Comandos
- Comandos Usados

Su tamaño, variará en función de lo que quiera almacenar el usuario, pero en principio se presupone que va a ser bastante grande puesto que los ficheros de *log* suelen contener miles y miles de registros que analizar (comandos usados) y cuantos más comandos se utilicen en la base del análisis (tabla comandos), mejores resultados se obtendrán, más reales, y fehacientes.

Veamos un ejemplo.

Para los ficheros que se proporcionan con el sistema tenemos que el fichero de comandos iniciales contiene un total de 2179 comandos con sus respectivas categorías y atributos: número de parámetros, dificultad, impacto y descripción. Y el fichero de *Logs* que queremos analizar, contiene un total de 289872 registros, con su fecha, usuario, comando y parámetros correspondientes.

Estos datos, serán almacenados en las tablas del sistema quedando el espacio repartido de la siguiente manera:

Tabla	Nº Filas	Tamaño (MB)
Comandos	2179	0,25
Categorías	8	0,06
Usuarios	32	0,19
Comandos Usados	289872	20,00
Query Reports	10	0,06
Tipo de Gráfico	73	0,06
Plantillas	4	0,06

Tabla 117: Tamaño de las tablas del sistema

Estos tamaños han sido sacados directamente de las estadísticas proporcionadas por el sistema gestor de base de datos, calculados de forma aproximada teniendo en cuenta el tipo de datos que se ha definido para cada uno de los campos de las tablas, el tamaño medio de las filas, el número de filas de la tabla y el tipo de almacenamiento propio del sistema gestor de base de datos que estamos utilizando (Oracle [23]).

Una vez analizados todos los aspectos críticos del diseño, se determina la conversión de las entidades en tablas, considerando las relaciones existentes entre ellas, sus identificadores y sus claves primarias y ajenas.

En la siguiente ilustración se muestra el modelo relacional que tendrá la base de datos del sistema que estamos diseñando, construido a partir del modelo entidad/relación determinado en el apartado 4.3.1 Modelo Conceptual de Datos y siguiendo las reglas de transformación especificadas para tal fin [31].

Como puede verse comparando ambos modelos, la transformación del modelo entidad/relación en el modelo relacional, ha consistido básicamente en la transformación de las entidades en tablas físicas de la base de datos, con el correspondiente formato de campos que puede verse en la imagen inferior; en la transformación de la relación 1:N entre categoría y comando en un atributo directo de la entidad débil dentro de la fuerte, y en el añadido de una tabla adicional (*commands_used*) para representar la relación M:N “usa” entre usuario y comando .

Además, se ha mantenido la entidad *Query_Reports*, para representar los datos de los distintos de informes que será capaz de realizar la aplicación, complementándola con las tablas *Report_Templates* y *Chart_Types*, para proporcionar más información acerca del tipo de gráfico que tendrá el informe, y la plantilla que se usará para generarlo.

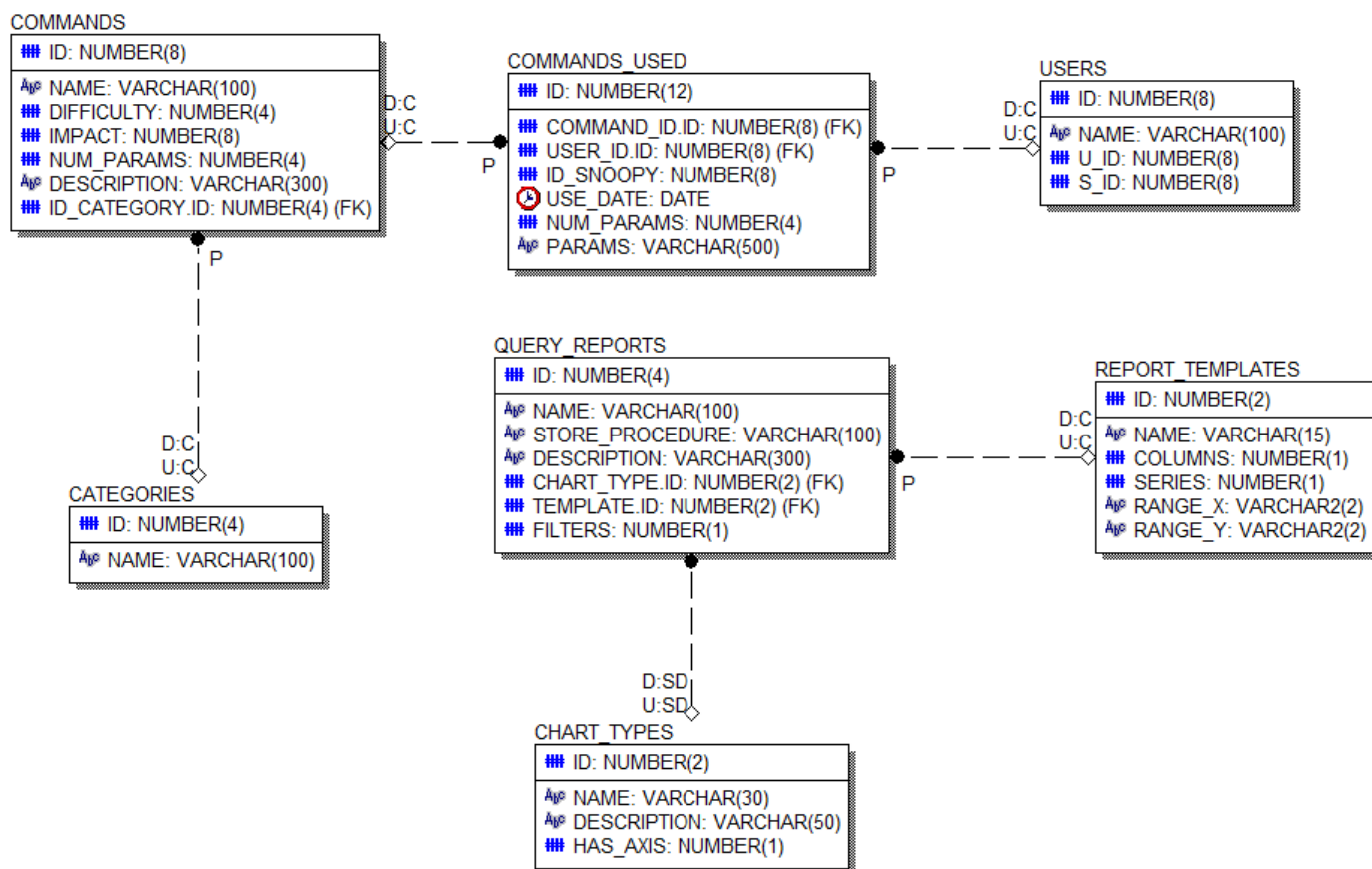


Ilustración 50: Modelo Relacional

5.5.2 Especificación de la Distribución de Datos

(Métrica v.3 [12] Tarea DSI 6.4)

El objetivo de esta tarea es determinar el modelo de distribución de los datos, teniendo en cuenta los requisitos de diseño especificados.

Para ello, se establece la ubicación del sistema gestor de base de datos y de los elementos de la estructura física de los datos, en los nodos correspondientes, de acuerdo a la arquitectura definida en el apartado 5.1 del presente documento (Definición de la Arquitectura del Sistema).

El sistema de información que estamos diseñando, consiste básicamente en el desarrollo de una aplicación pesada que ayude al análisis de los registros almacenados en un fichero de *logs* determinado por el usuario. Esto, hace que la distribución de los datos pueda ser diversa, en función de las necesidades del momento. Esto es, podemos colocar el sistema gestor de la base de datos tanto en el mismo equipo donde esté instalada la aplicación, como en una máquina o servidor distinto, haciendo que la aplicación se conecte directamente a él a través de su configuración.

5.6 Generación de Especificaciones de Construcción

Este apartado se corresponde con la Actividad DSI 8 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta actividad consiste básicamente en determinar aquellas especificaciones que tenemos que tener en cuenta para la construcción del sistema de información diseñado en los apartados anteriores. Dichas especificaciones hay que definir las de tal forma, que el programador del sistema solo tenga que traducir cada componente al lenguaje de programación seleccionado (entendiendo por componente aquellas unidades independientes y coherentes de construcción y ejecución, que se corresponden con un empaquetamiento físico de los elementos del diseño de detalle, como pueden ser los módulos, las clases o las especificaciones de las interfaces).

En esta actividad, también se definirá el orden o secuencia que se debe seguir en la construcción de los componentes especificados y en la realización de las pruebas correspondientes.

5.6.1 Especificación del Entorno de Construcción

(Métrica v.3 [12] Tarea DSI 8.1)

Esta tarea tiene como objetivo principal la definición detallada y completa del entorno imprescindible para la construcción de los componentes del sistema de información.

- **Entorno tecnológico:** hardware y software

Para la construcción del sistema de información que estamos diseñando, según las especificaciones marcadas en el diseño detallado elaborado en los apartados anteriores del presente documento, se requerirá de un equipo con las siguientes prestaciones mínimas:

Hardware:

- ✓ **Procesador** 1.6 GHz de CPU.
- ✓ **Memoria RAM** de 1 GB.
- ✓ **Disco duro** de 25 GB como mínimo (para todos los requisitos SW).
- ✓ **Red de comunicaciones:** equipo conectado a una red con salida para el envío de correos electrónicos.
- ✓ **Tarjeta de vídeo y monitor** SVGA, con una resolución de 1024x768.
- ✓ **Ratón y Teclado** compatible.

Software:

- ✓ **Sistema Operativo:** Windows 7.
 - ✓ **Base de datos:** Oracle 10g Express Edition [23]
 - ✓ **Lenguaje de Programación y Entorno de Desarrollo:** C#.Net [17] y Visual Studio 2008 [18]
-
- **Herramientas para la generación de Informes y documentación:** Acrobat Reader PDF, Enterprise Architect 6.5, ERwin Data Modeler, y el paquete completo de Office 2003 con las herramientas de Microsoft Word, Project, Power Point y Excel.

5.6.2 Elaboración de las Especificaciones del Modelo Físico de Datos

(Métrica v.3 [12] Tarea DSI 8.4)

Esta tarea tiene como fin principal especificar de forma detallada cada componente del modelo físico de datos descrito en el apartado 5.5 del presente documento (Diseño Físico de Datos).

Dicha especificación consistirá, por un lado, en describir la creación de las diferentes tablas que utilizaremos para el almacenamiento de los datos, incorporando las restricciones del entorno establecidas anteriormente sobre cada una de ellas; y por otro, en especificar las órdenes de inserción de la carga inicial de datos necesaria para el correcto funcionamiento del sistema y la creación de las estructuras de almacenamiento indispensable para su ejecución.

A) Creación de las Estructuras de Almacenamiento (Tablas)

Tablas para el Almacenamiento de Datos

1. Tabla de Categorías

```
DROP TABLE CATEGORIES CASCADE CONSTRAINTS ;

CREATE TABLE CATEGORIES
(
    ID                NUMBER(4) NOT NULL ,
    NAME              VARCHAR2(100) NOT NULL ,
    PRIMARY KEY (ID)
);
```

2. Tabla de Comandos

```
DROP TABLE COMMANDS CASCADE CONSTRAINTS;

CREATE TABLE COMMANDS
(
    ID                NUMBER(8) NOT NULL,
    NAME              VARCHAR2(100) NOT NULL,
    DIFFICULTY        NUMBER(4) NOT NULL,
    IMPACT            NUMBER(8) NOT NULL,
    CATEGORY_ID       NUMBER(4) NOT NULL,
    NUM_PARAMS        NUMBER(4) NOT NULL,
    DESCRIPTION        VARCHAR2(300) NOT NULL,
    PRIMARY KEY (ID),
    FOREIGN KEY (CATEGORY_ID)
                    REFERENCES CATEGORIES (ID)
                    ON DELETE CASCADE
);
```

3. Tabla de Usuarios

```
DROP TABLE USERS CASCADE CONSTRAINTS;

CREATE TABLE USERS
(
    ID                NUMBER(8) NOT NULL,
    NAME              VARCHAR2(100) NOT NULL,
    U_ID              NUMBER(8) NOT NULL,
    S_ID              NUMBER(8) NOT NULL,
    PRIMARY KEY (ID)
);
```

4. Tabla de Comandos Usados o Logs

```
DROP TABLE COMMANDS_USED CASCADE CONSTRAINTS;

CREATE TABLE COMMANDS_USED
(
    ID                NUMBER(12) NOT NULL,
    COMMAND_ID        NUMBER(8) NOT NULL,
    USER_ID           NUMBER(8) NOT NULL,
    ID_SNOOPY         NUMBER(8) NOT NULL,
    USE_DATE          DATE NOT NULL,
    NUM_PARAMS        NUMBER(4) NOT NULL,
    PARAMS            VARCHAR2(500),
    PRIMARY KEY (ID),
    FOREIGN KEY (COMMAND_ID)
                    REFERENCES COMMANDS (ID)
                    ON DELETE CASCADE,
    FOREIGN KEY (USER_ID)
                    REFERENCES USERS (ID)
                    ON DELETE CASCADE
);
```

Tablas para la Realización de Informes

5. Tabla de Informes

```
DROP TABLE QUERY_REPORTS CASCADE CONSTRAINTS;

CREATE TABLE QUERY_REPORTS
(
    ID                NUMBER(4) NOT NULL,
    NAME              VARCHAR2(100) NOT NULL,
    STORE_PROCEDURE   VARCHAR2(100) NOT NULL,
    DESCRIPTION        VARCHAR2(300) NOT NULL,
    CHART_TYPE         NUMBER(2) NOT NULL,
    TEMPLATE           NUMBER(2) NOT NULL,
    FILTERS            FILTER(1) NOT NULL,
    PRIMARY KEY (ID),
    UNIQUE KEY (NAME),
    FOREIGN KEY (CHART_TYPE)
        REFERENCES CHAR_TYPES (ID)
        ON DELETE CASCADE,
    FOREIGN KEY (TEMPLATE)
        REFERENCES REPORT_TEMPLATES(ID)
        ON DELETE CASCADE
);
```

6. Tabla de Tipos de Gráficos Válidos

```
CREATE TABLE CHART_TYPES
(
    ID                NUMBER(2) NOT NULL,
    NAME              VARCHAR2(50) NOT NULL,
    HAS_AXIS          NUMBER (1) NOT NULL,
    PRIMARY KEY (ID),
    UNIQUE KEY (NAME)
);
```

7. Tabla de Plantillas para la elaboración de informes

```
CREATE TABLE REPORT_TEMPLATES
(
    ID                NUMBER(2) NOT NULL,
    NAME              VARCHAR2(15) NOT NULL,
    COLUMNS          NUMBER (1) NOT NULL,
    SERIES            NUMBER (1) NOT NULL,
    RANGE_X           VARCHAR2(2) NOT NULL,
    RANGE_Y           VARCHAR2(2) NOT NULL,
    PRIMARY KEY (ID),
    UNIQUE KEY (NAME)
);
```

B) Creación de las Secuencias y PL's específicos del funcionamiento del sistema:

Tabla	Secuencia	PL Inserción Registros	PL Reinicio Secuencias
COMMANDS	SEQ_COMMANDS	PL_INSERT_COMMANDS PL_INSERT_COMMANDS_INICIAL	PL_RESTART_ALL_SEQUENCES (Reinicia las 4 secuencias)
CATEGORIES	SEQ_CATEGORIES	PL_INSERT_CATEGORIES	PL_RESTART_BOTH_SEQUENCES (Reinicia Comandos y Categorías) PL_RESTART_SEQUENCES (Reinicia Comandos Usados y Usuarios)
USERS	SEQ_USERS	PL_INSERT_USERS	
COMMANDS_USED	SEQ_COMMANDS_USED	PL_INSERT_USED_COMMANDS	

Tabla 118: PL's y Secuencias específicos del sistema

Ejemplo de código para la creación de una secuencia:

```
CREATE SEQUENCE SEQ_CATEGORIES
MINVALUE 0
MAXVALUE 9999
START WITH 0
INCREMENT BY 1
CACHE 20 ;
```

Utilizaremos las secuencias creadas para generar la clave principal de los registros insertados (ID) de cada una de las tablas de la base de datos, de manera que consigamos hacerlos consecutivos y únicos.

Ejemplo de código para la creación de un PL de inserción de registros

```
CREATE OR REPLACE PROCEDURE PL_INSERT_CATEGORIES
(
    P_NAME          IN CATEGORIES.NAME%TYPE
)
IS
    V_COUNTER NUMBER;

BEGIN

    SELECT COUNT(*) INTO V_COUNTER
    FROM CATEGORIES C
    WHERE C.NAME = P_NAME;

    IF V_COUNTER=0 THEN

        INSERT INTO CATEGORIES (ID, NAME)
        VALUES (SEQ_CATEGORIES.NEXTVAL, P_NAME);
        COMMIT;

    END IF;

END PL_INSERT_CATEGORIES;
```

Utilizaremos los PL's de inserción de registros para poder introducir una lista de objetos a la vez, y así minimizar el número de accesos a la base de datos y agilizar el tiempo de carga de datos de la aplicación.

C) Inserción de todos los tipos de registros que van a poder realizarse con el sistema, y sus PL's correspondientes.

- Se insertará un registro por cada tipo de informe a generar y un PL con la QUERY que contiene la lógica de análisis del informe.

Ejemplo de INSERT para la tabla QUERY_REPORT

```
INSERT INTO QUERY_REPORT
  (ID, NAME, STORE_PROCEDURE, DESCRIPTION)
VALUES (1,
        'MOST USED COMMANDS',
        'PL_MOST_USED_COMMANDS',
        'INFORME DE LOS COMANDOS MÁS USADOS');
```

Ejemplo de PL de ejecución de QUERY

```
CREATE OR REPLACE PROCEDURE PL_MOST_USED_COMMANDS
(
  P_REF OUT SYS_REFCURSOR
)
IS
BEGIN
  OPEN P_REF FOR

  -- SELECT * FROM COMMANDS;

  SELECT C.NAME, COUNT(COMMAND_ID) as VECES_USADO
  FROM COMMANDS_USED U, COMMANDS C
  WHERE U.COMMAND_ID = C.ID;
  GROUP BY C.NAME
  ORDER BY VECES_USADO DESC;

END PL_MOST_USED_COMMANDS;
```

Esta tabla, solo podrá ser cargada y modificada antes del uso de la aplicación, nunca, durante la ejecución de la misma, ya que toda la lógica de análisis del sistema y por tanto, la funcionalidad principal de la aplicación, gira en torno a la correcta cumplimentación de esta tabla, y sin una tabla Query_Reports correctamente cumplimentada, no será posible conseguir el funcionamiento deseado del sistema.

El usuario cuyo perfil concuerde con el descrito por nosotros como “usuario administrador” (es decir, aquel que estuviese interesado en alterar la lógica de análisis del sistema, los tipos de informes a realizar, y/o la base de análisis de la aplicación) será quien pueda realizar los cambios y/o actualizaciones oportunas en esta tabla, según sus necesidades. Para todos los demás, esta tabla será siempre la misma, es decir, la proporcionada por el sistema de origen, por lo que le bastará con seguir correctamente las indicaciones de carga e instalación de la aplicación para conseguir arrancar el sistema sin ningún tipo de problema.

D) Orden correcto para el borrado de las Tablas

1. `DELETE FROM COMMANDS_USED;` 2. `DELETE FROM COMMANDS;`
3. `DELETE FROM CATEGORIES;` 4. `DELETE FROM USERS.`

Las tablas referentes a los informes, no deberían borrarse nunca, pues sin ellas el sistema no será capaz de generar ningún análisis. No obstante, si el usuario quisiera borrarlas, el orden correcto de borrado de las mismas sería:

1. `DELETE FROM REPORT_TEMPLATES;` 2. `DELETE FROM CHART_TYPE;`
3. `DELETE FROM QUERY_REPORTS.`

5.7 Especificación Técnica del Plan de Pruebas

Este apartado se corresponde con la Actividad DSI 10 del proceso de Diseño del sistema de Información [DSI] de Métrica v.3 [12].

Esta actividad tiene como objetivo principal la especificación en detalle del plan de pruebas que se va a seguir para comprobar el correcto funcionamiento del sistema.

Se especificarán pruebas de distintos niveles, distinguiéndose cuatro grupos principales:

- Pruebas unitarias: Tienen como objetivo principal verificar la funcionalidad y estructura de cada componente del sistema de información de forma individual.
- Pruebas de integración: Tienen por objetivo verificar el correcto ensamblaje entre los distintos componentes de un mismo subsistema.
- Pruebas del sistema: Tienen como objetivo principal la verificación del correcto funcionamiento del sistema en su conjunto.
- Pruebas de aceptación: Tienen como objetivo verificar que el sistema cumple con todas las funcionalidades especificadas por el cliente.

El punto de partida para la especificación de las pruebas a realizar serán el catálogo de casos de usos, el catálogo de requisitos, el catálogo de excepciones y el diseño detallado, realizados en actividades anteriores; asegurándonos así que el sistema cumple con todos los requisitos planteados por el cliente/usuario.

Para la especificación de las pruebas del catálogo, se utilizará el siguiente formato en tabla:

Identificador	PR_T_XY
Nombre	<i>Nombre de la prueba</i>
Descripción	<i>Descripción genérica de qué es lo que se quiere conseguir con la prueba.</i>
Entrada	<i>Qué hay que hacer para realizar la prueba</i>
Salida	<i>Qué respuesta se espera obtener con esta prueba</i>
Requisitos del Entorno	<i>Requisitos indispensables que se deben cumplir para la correcta realización de la prueba</i>

Tabla 119: Formato General para el catálogo de Pruebas

Donde, el identificador estará compuesto por:

- PR: para indicar que es una prueba
- T: para especificar el tipo de prueba que se está describiendo
 - o UN – prueba unitaria
 - o IN – prueba de integración
 - o SI – prueba de sistema
 - o AC – prueba de aceptación
- XY: son dos cifras que actuarán como contador para numerar las distintas pruebas de cada uno de los grupos del catálogo.

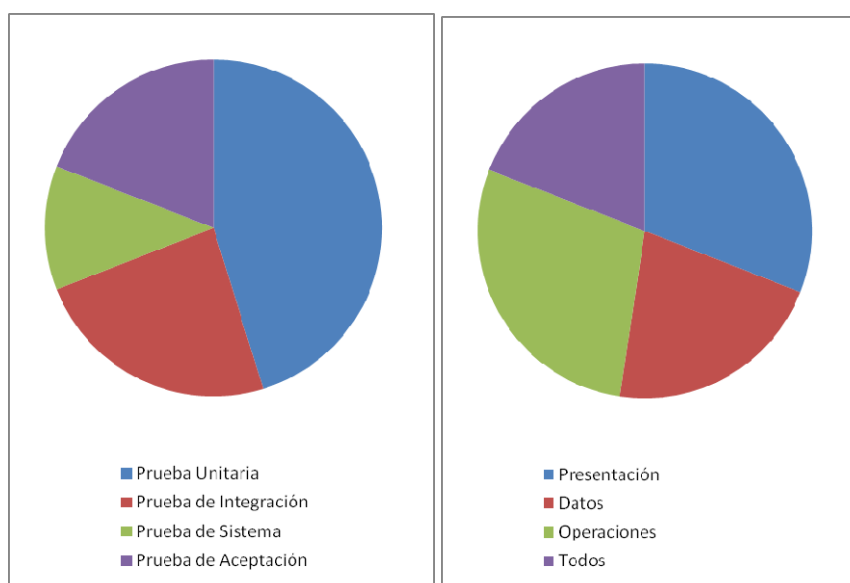


Ilustración 51: Diagrama Distribución de Pruebas por Tipos y Módulos

- **Catálogo de Pruebas Unitarias:**

Pruebas que se encargan de verificar cada componente del sistema de información, comprobando la funcionalidad y estructura de cada componente de forma individual.

Identificador	PR-UN-01
Nombre	Borrado de base de datos cuando tiene datos.
Descripción	Realizar un borrado de la base de datos cuando ésta contenga datos de un análisis de logs anterior.
Entrada	Pulsar el botón "Borrar datos almacenados" cuando haya datos almacenados de un análisis de logs realizado previamente.
Salida	El sistema deberá borrar correctamente las tablas de logs e informar del resultado de la operación.
Requisitos del Entorno	N/A
Módulo afectado	Datos.

Tabla 120: Prueba Unitaria 01 (PR_UN_01)

Identificador	PR-UN-02
Nombre	Carga de comandos iniciales con un fichero incorrecto.
Descripción	Cargar un fichero de comandos iniciales con un formato incorrecto, es decir, que no sea una Excel.
Entrada	Fichero con formato no válido.
Salida	Se mostrará un mensaje de error.
Requisitos del Entorno	N/A
Módulo afectado	Operaciones.

Tabla 121: Prueba Unitaria 02 (PR_UN_02)

Identificador	PR-UN-03
Nombre	Carga de comandos iniciales con un fichero vacío.
Descripción	Cargar un fichero de comandos iniciales con un fichero Excel vacío.
Entrada	Fichero Excel sin datos.
Salida	Se mostrará un mensaje de error.
Requisitos del Entorno	N/A
Módulo afectado	Operaciones.

Tabla 122: Prueba Unitaria 03 (PR_UN_03)

Identificador	PR-UN-04
Nombre	<i>Carga de comandos iniciales con un fichero con datos no válidos.</i>
Descripción	<i>Cargar un fichero de comandos iniciales con un fichero Excel que contenga algún dato inválido.</i>
Entrada	<i>Fichero Excel con datos no válidos.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 123: Prueba Unitaria 04 (PR_UN_04)

Identificador	PR-UN-05
Nombre	<i>Carga de un fichero de logs con formato incorrecto.</i>
Descripción	<i>Cargar un fichero de logs con un formato incorrecto (formato Snoopy no admitido o directamente otro formato).</i>
Entrada	<i>Fichero de logs con formato no válido.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 124: Prueba Unitaria 05 (PR_UN_05)

Identificador	PR-UN-06
Nombre	<i>Carga de un fichero de logs con un fichero vacío.</i>
Descripción	<i>Cargar un fichero de logs con un fichero vacío.</i>
Entrada	<i>Fichero de logs vacío.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 125: Prueba Unitaria 06 (PR_UN_06)

Identificador	PR-UN-07
Nombre	<i>Carga de un fichero de logs con datos no válidos.</i>
Descripción	<i>Cargar un fichero de logs que contiene algún dato no válido.</i>
Entrada	<i>Fichero de logs con datos no válidos.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 126: Prueba Unitaria 07 (PR_UN_07)

Identificador	PR-UN-08
Nombre	<i>Analizar datos almacenados cuando la base de datos no tiene ningún fichero de logs cargado.</i>
Descripción	<i>Realizar un análisis de datos, sin haber cargado previamente algún fichero de logs correctamente en la aplicación.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", sin datos en el SGBD para analizar.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 127: Prueba Unitaria 08 (PR_UN_08)

Identificador	PR-UN-09
Nombre	<i>Obtener un informe sin seleccionar los tipos de informe a obtener.</i>
Descripción	<i>Generar un informe sin haber añadido previamente ningún informe al listado de informes a generar.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", seleccionar un filtro y pulsar el botón "Generar Informe".</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Presentación.</i>

Tabla 128: Prueba Unitaria 09 (PR_UN_09)

Identificador	PR-UN-10
Nombre	<i>Reiniciar los filtros de la pantalla de selección de informes.</i>
Descripción	<i>Reiniciar los filtros de la pantalla de selección de informes, habiendo filtrado previamente los datos del formulario por alguno de los tipos posible.</i>
Entrada	<i>Pulsar el botón de restear filtros.</i>
Salida	<i>Se borrarán todos los filtros actuales en la pantalla de generación de informes.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 129: Prueba Unitaria 10 (PR_UN_10)

Identificador	PR-UN-11
Nombre	<i>Generar informes cuando los ficheros de la generación anterior están abiertos.</i>
Descripción	<i>Generar informes en la aplicación, estando abierto alguno de los ficheros que se generaron en una ejecución previa de la generación de informes.</i>
Entrada	<i>Pulsar el botón de generación de informes.</i>
Salida	<i>Se mostrará un mensaje de error, avisando de que es necesario cerrar previamente dichos ficheros.</i>
Requisitos del Entorno	<i>Ficheros de una generación de informes previa abiertos.</i>
Módulo afectado	<i>Presentación.</i>

Tabla 130: Prueba Unitaria 11 (PR_UN_11)

Identificador	PR-UN-12
Nombre	<i>Obtener un informe por email, sin rellenar el correo electrónico.</i>
Descripción	<i>Obtener un informe generado enviándolo por email. Dejar vacío el campo de la dirección de correo electrónico de destino.</i>
Entrada	<i>En la pantalla de resultados del análisis, pulsar el botón de envío por email sin rellenar el correo electrónico al que se enviarán los informes.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 131: Prueba Unitaria 12 (PR_UN_12)

Identificador	PR-UN-13
Nombre	<i>Obtener un informe por email, con una dirección de correo electrónico no válida.</i>
Descripción	<i>Obtener un informe generado enviándolo por email. Introducir una dirección de correo electrónico con un formato incorrecto.</i>
Entrada	<i>En la pantalla de resultados del análisis, pulsar el botón de envío por email introduciendo un email con formato no válido.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 132: Prueba Unitaria 13 (PR_UN_13)

Identificador	PR-UN-14
Nombre	<i>Realizar otro informe desde la pantalla de resultados de la aplicación.</i>
Descripción	<i>Una vez generados correctamente uno o varios informes, volver a la pantalla de generación de informes para seguir generando otros informes.</i>
Entrada	<i>Pulsar el botón "Realizar otro informe".</i>
Salida	<i>Se volverá a la pantalla de generación de informes, manteniendo la selección del usuario.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 133: Prueba Unitaria 14 (PR_UN_14)

Identificador	PR-UN-15
Nombre	<i>Volver al menú principal, desde la pantalla de resultados de la aplicación.</i>
Descripción	<i>Una vez generados correctamente uno o varios informes, volver al menú principal de la aplicación.</i>
Entrada	<i>Pulsar el botón "Volver al menú principal".</i>
Salida	<i>Se volverá a la pantalla principal de la aplicación, borrando la información de filtrado e informes de la aplicación.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 134: Prueba Unitaria 15 (PR_UN_15)

Identificador	PR-UN-16
Nombre	<i>Cambiar el idioma de la aplicación a "Inglés".</i>
Descripción	<i>En la pantalla principal de la aplicación, cambiar el idioma a "Inglés".</i>
Entrada	<i>Pulsar el icono con la bandera de UK.</i>
Salida	<i>Se cambiará el idioma para toda la aplicación en esta sesión a "Inglés".</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 135: Prueba Unitaria 16 (PR_UN_16)

Identificador	PR-UN-17
Nombre	<i>Cambiar el idioma de la aplicación a "Español".</i>
Descripción	<i>En la pantalla principal de la aplicación, cambiar el idioma a "Español".</i>
Entrada	<i>Pulsar el icono con la bandera de ES.</i>
Salida	<i>Se cambiará el idioma para toda la aplicación en esta sesión a "Español".</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 136: Prueba Unitaria 17 (PR_UN_17)

Identificador	PR-UN-18
Nombre	<i>Abrir la sección "Guía de uso" del menú de la aplicación.</i>
Descripción	<i>En la pantalla principal de la aplicación, abrir la guía de uso de la aplicación a través del menú.</i>
Entrada	<i>Pulsar el submenú "Guía de uso" del menú de "Información".</i>
Salida	<i>Se abrirá un documento con una guía de uso paso a paso de la aplicación.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 137: Prueba Unitaria 18 (PR_UN_18)

Identificador	PR-UN-19
Nombre	<i>Abrir la sección "Acerca de..." del menú de la aplicación.</i>
Descripción	<i>En la pantalla principal de la aplicación, abrir la sección "Acerca de..." a través del menú.</i>
Entrada	<i>Pulsar el submenú "Acerca de..." del menú de "Información".</i>
Salida	<i>Se abrirá una ventana con la información sobre la creación de la propia aplicación.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Presentación.</i>

Tabla 138: Prueba Unitaria 19 (PR_UN_19)

▪ **Catálogo de Pruebas de Integración:**

Pruebas que se encargan de las verificaciones asociadas a los grupos de componentes que constituyen los subsistemas, verificando el correcto ensamblaje entre los distintos componentes.

Identificador	PR-IN-01
Nombre	<i>Borrado de base de datos cuando ya está vacía.</i>
Descripción	<i>Realizar un borrado de los datos almacenados cuando no haya ningún dato almacenado.</i>
Entrada	<i>Pulsar el botón "Borrar datos almacenados" cuando no hay nada que borrar.</i>
Salida	<i>El sistema intentará borrar los datos y al no haber nada que borrar informará que la tarea de borrado ha finalizado correctamente.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Datos.</i>

Tabla 139: Prueba de Integración 01 (PR_IN_01)

Identificador	PR-IN-02
Nombre	<i>Carga de comandos iniciales cuando la base de datos está vacía.</i>
Descripción	<i>Cargar un fichero de comandos iniciales, estando la base de datos vacía.</i>
Entrada	<i>Fichero Excel de comandos correcto.</i>
Salida	<i>Se importarán correctamente los comandos del fichero de entrada.</i>
Requisitos del Entorno	<i>Base de datos vacía.</i>
Módulo afectado	<i>Datos.</i>

Tabla 140: Prueba de Integración 02 (PR_IN_02)

Identificador	PR-IN-03
Nombre	<i>Carga de comandos iniciales cuando la base de datos ya tiene datos de comandos y/o logs insertados.</i>
Descripción	<i>Cargar un fichero de comandos iniciales, cuando ya existen comandos y logs insertados.</i>
Entrada	<i>Fichero Excel de comandos correcto.</i>
Salida	<i>Se importarán los comandos, borrando previamente los comandos y logs que hubiera almacenados.</i>
Requisitos del Entorno	<i>Base de datos con información sobre comandos y/o logs.</i>
Módulo afectado	<i>Datos.</i>

Tabla 141: Prueba de Integración 03 (PR_IN_03)

Identificador	PR-IN-04
Nombre	<i>Carga de un fichero de logs cuando la base de datos está vacía.</i>
Descripción	<i>Cargar un fichero de logs cuando las tablas de comandos y logs están vacías.</i>
Entrada	<i>Fichero de logs correcto.</i>
Salida	<i>La aplicación informará al usuario que previamente es necesario cargar la base de comandos.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Datos.</i>

Tabla 142: Prueba de Integración 04 (PR_IN_04)

Identificador	PR-IN-05
Nombre	<i>Carga de un fichero de logs cuando la BBDD ya tiene la base de comandos insertada correctamente.</i>
Descripción	<i>Cargar un fichero de logs cuando ya están cargados correctamente los comandos y categorías.</i>
Entrada	<i>Fichero de logs correcto.</i>
Salida	<i>Se deberán insertar los logs correctamente.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 143: Prueba de Integración 05 (PR_IN_05)

Identificador	PR-IN-06
Nombre	<i>Analizar datos, filtrando los informes por uno de los tipos posibles.</i>
Descripción	<i>Realizar un análisis de datos, filtrando los informes por una de las categorías posibles.</i>
Entrada	<i>Seleccionar un tipo de filtro en la pantalla de generación de informes (Todos, Usuarios, Comandos, Categorías o Generales).</i>
Salida	<i>Se mostrará la segunda parte del formulario de generación de informes, con los tipos de informe y de filtrado disponibles para cada uno de ellos.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Presentación.</i>

Tabla 144: Prueba de Integración 06 (PR_IN_06)

Identificador	PR-IN-07
Nombre	<i>Intentar obtener un informe cuando no hay ningún tipo de informe en BBDD.</i>
Descripción	<i>Realizar un análisis de datos cuando no existe ningún tipo de informe cargado en BBDD.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", sin informes en el SGBD.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>La tabla de informes deberá estar vacía.</i>
Módulo afectado	<i>Presentación.</i>

Tabla 145: Prueba de Integración 07 (PR_IN_07)

Identificador	PR-IN-08
Nombre	<i>Filtrar un informe seleccionado que admita filtro de usuario.</i>
Descripción	<i>Seleccionar un usuario como filtro en alguno de los informes de la aplicación que admitan este tipo de filtro.</i>
Entrada	<i>Usuario seleccionado en el panel de filtros.</i>
Salida	<i>Se actualizarán los selectores del resto de filtros, de acuerdo a la selección del usuario, y se guardará el filtro para los informes que estén seleccionados.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Operaciones.</i>

Tabla 146: Prueba de Integración 08 (PR_IN_08)

Identificador	PR-IN-09
Nombre	<i>Filtrar un informe seleccionado que admita filtro de comando.</i>
Descripción	<i>Seleccionar un comando como filtro en alguno de los informes de la aplicación que admitan este tipo de filtro.</i>
Entrada	<i>Comando seleccionado en el panel de filtros.</i>
Salida	<i>Se actualizarán los selectores del resto de filtros, de acuerdo a la selección del usuario, y se guardará el filtro para los informes que estén seleccionados.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Operaciones.</i>

Tabla 147: Prueba de Integración 09 (PR_IN_09)

Identificador	PR-IN-10
Nombre	<i>Filtrar un informe seleccionado que admita filtro de categoría.</i>
Descripción	<i>Seleccionar una categoría como filtro en alguno de los informes de la aplicación que admitan este tipo de filtro.</i>
Entrada	<i>Categoría seleccionada en el panel de filtros.</i>
Salida	<i>Se actualizarán los selectores del resto de filtros, de acuerdo a la selección del usuario, y se guardará el filtro para los informes que estén seleccionados.</i>
Requisitos del Entorno	N/A
Módulo afectado	<i>Operaciones.</i>

Tabla 148: Prueba de Integración 10 (PR_IN_10)

▪ **Catálogo de Pruebas del Sistema:**

Pruebas de integración del sistema de información completo. Prueban el sistema en su conjunto para verificar que las especificaciones técnicas y funcionales se cumplen.

Identificador	PR-SI-01
Nombre	<i>Borrado de base de datos cuando no hay conexión con el SGBD.</i>
Descripción	<i>Realizar un borrado de los datos almacenados en la base de datos cuando el SGBD no se encuentra disponible.</i>
Entrada	<i>Parar la ejecución del SGBD, y a continuación pulsar el botón "Borrar datos almacenados".</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>Parar previamente el SGBD.</i>
Módulo afectado	<i>Datos.</i>

Tabla 149: Prueba de Sistema 01 (PR_SI_01)

Identificador	PR-SI-02
Nombre	<i>Carga de comandos iniciales cuando no hay conexión con el SGBD.</i>
Descripción	<i>Cargar un fichero de comandos iniciales cuando el SGBD no se encuentra disponible.</i>
Entrada	<i>Parar la ejecución del SGBD, y a continuación pulsar el botón "Cargar comandos".</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>Parar previamente el SGBD.</i>
Módulo afectado	<i>Datos.</i>

Tabla 150: Prueba de Sistema 02 (PR_SI_02)

Identificador	PR-SI-03
Nombre	<i>Carga de un fichero de logs cuando no hay conexión con el SGBD.</i>
Descripción	<i>Cargar un fichero de logs cuando el SGBD no se encuentra disponible.</i>
Entrada	<i>Parar la ejecución del SGBD, y a continuación pulsar el botón "Cargar fichero log".</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>Parar previamente el SGBD.</i>
Módulo afectado	<i>Datos.</i>

Tabla 151: Prueba de Sistema 03 (PR_SI_03)

Identificador	PR-SI-04
Nombre	<i>Analizar datos almacenados cuando no hay conexión con el SGBD.</i>
Descripción	<i>Realizar un análisis de los datos importados previamente en la BBDD, pero estando el SGBD no disponible.</i>
Entrada	<i>Parar la ejecución del SGBD, y a continuación pulsar el botón "Analizar Datos".</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>Parar previamente el SGBD.</i>
Módulo afectado	<i>Datos.</i>

Tabla 152: Prueba de Sistema 04 (PR_SI_04)

Identificador	PR-SI-05
Nombre	<i>Obtener un informe por email, sin conexión a Internet.</i>
Descripción	<i>Obtener un informe generado enviándolo por email, sin tener conexión a Internet en el momento del envío.</i>
Entrada	<i>Enviar informes por email.</i>
Salida	<i>Se mostrará un mensaje de error.</i>
Requisitos del Entorno	<i>Desactivar o deshabilitar la conexión a Internet en el PC.</i>
Módulo afectado	<i>Operaciones.</i>

Tabla 153: Prueba de Sistema 05 (PR_SI_05)

- **Catálogos de Pruebas de Aceptación:**

Pruebas que se encargan de validar que el sistema cumple con los requisitos de funcionamiento esperado. Se basa en el catálogo de requisitos, de manera que busca conseguir la aceptación final del sistema por parte del cliente/usuario.

Identificador	PR-AC-01
Nombre	<i>Carga de comandos iniciales correcta.</i>
Descripción	<i>Cargar un fichero de comandos válido.</i>
Entrada	<i>Fichero Excel de comandos correcto.</i>
Salida	<i>Se importarán correctamente los comandos del fichero de entrada.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 154: Prueba de Aceptación 01 (PR_AC_01)

Identificador	PR-AC-02
Nombre	<i>Carga de un fichero de logs correcto.</i>
Descripción	<i>Cargar un fichero de logs correcto.</i>
Entrada	<i>Fichero de logs correcto.</i>
Salida	<i>Se deberán insertar los logs correctamente, borrando previamente los posibles logs que hubiera en la tabla de logs.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 155: Prueba de Aceptación 02 (PR_AC_02)

Identificador	PR-AC-03
Nombre	<i>Analizar datos correctamente.</i>
Descripción	<i>Realizar un análisis de datos, habiendo cargado antes los comandos y logs, y teniendo informes correctos en la BBDD.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", con datos para analizar.</i>
Salida	<i>Se mostrará el formulario de generación de informes.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 156: Prueba de Aceptación 03 (PR_AC_03)

Identificador	PR-AC-04
Nombre	<i>Obtener un informe seleccionando un solo tipo de informe de los posibles.</i>
Descripción	<i>Generar un único informe de entre los posibles que muestre la aplicación.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", seleccionar un filtro, añadir un tipo de informe y pulsar en el botón de generación de informes.</i>
Salida	<i>Se mostrará el formulario de obtención de informes.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 157: Prueba de Aceptación 04 (PR_AC_04)

Identificador	PR-AC-05
Nombre	<i>Obtener un informe seleccionando varios tipos de informe de entre los posibles.</i>
Descripción	<i>Generar varios tipos de informe a la vez, de entre los posibles que muestre la aplicación.</i>
Entrada	<i>Pulsar el botón "Analizar Datos", seleccionar un filtro, añadir los tipos de informes deseados y pulsar el botón de generación de informes.</i>
Salida	<i>Se mostrará el formulario de obtención de informes.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 158: Prueba de Aceptación 05 (PR_AC_05)

Identificador	PR-AC-06
Nombre	<i>Obtener informe en formato Excel.</i>
Descripción	<i>Obtener un informe generado en formato Excel.</i>
Entrada	<i>En la pantalla de resultados del análisis, pulsar el botón de obtención de los resultados en formato Excel.</i>
Salida	<i>Se abrirá el informe en formato Excel, permitiendo al usuario ver los resultados y guardar una copia de los mismos.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 159: Prueba de Aceptación 06 (PR_AC_06)

Identificador	PR-AC-07
Nombre	<i>Obtener informe en formato pdf.</i>
Descripción	<i>Obtener un informe generado en formato pdf.</i>
Entrada	<i>En la pantalla de resultados del análisis, pulsar el botón de obtención de los resultados en formato pdf.</i>
Salida	<i>Se abrirá el informe en formato pdf, permitiendo al usuario ver los resultados y guardar una copia de los mismos.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 160: Prueba de Aceptación 07 (PR_AC_07)

Identificador	PR-AC-08
Nombre	<i>Obtener un informe por email (Excel y/o pdf) correctamente.</i>
Descripción	<i>Obtener un informe generado enviándolo por email en formato Excel y/o PDF.</i>
Entrada	<i>En la pantalla de resultados del análisis, pulsar el botón de envío por email en alguno de los formatos disponibles, habiendo introducido previamente una dirección de correo válida.</i>
Salida	<i>Se enviará un correo electrónico con los informes en los formatos especificados como archivos adjuntos.</i>
Requisitos del Entorno	<i>N/A</i>
Módulo afectado	<i>Todos.</i>

Tabla 161: Prueba de Aceptación 08 (PR_AC_08)

6. Implementación

Este apartado se ha cumplimentado únicamente con el fin de poder exponer los detalles de implementación que queremos destacar, bien porque nos parecen los más interesantes de la aplicación, o bien porque han sido los que más nos ha costado realizar.

El código como tal no se incluirá directamente en la memoria, pero sí que se proporcionará un repositorio web donde se podrán encontrar todos los códigos fuente del sistema.

Repositorio web: <https://github.com/100052610/PFC>

Como detalles más interesantes para exponer de la implementación del sistema, cabe destacar principalmente los siguientes:

1. Creación del fichero de Comandos Iniciales
2. Lectura de Fichero y Almacenamiento de los Datos
3. Uso de PL's para la inserción masiva de registros en la base de datos del sistema
4. Elección del driver de conexión para enlazar la aplicación con el sistema gestor de base de datos
5. Uso de una librería para la conexión a la base de datos, el envío de correo electrónico y la generación de los *logs* propios de la ejecución del sistema.
6. Implementación de un patrón *observer* mediante la clase *BackgroundWorker* para informar del progreso de las tareas en la interfaz.
7. Inclusión de una librería para internacionalizar los textos de los mensajes de las ventanas de aviso de la interfaz.
8. Gestión de la navegación entre formularios utilizando el patrón Singleton.
9. Utilización de la librería de manejo de ficheros Excel para la generación de los informes.
10. Intento fallido de conexión directa con Google Drive.

Veamos cada una de ellas.

✓ Creación del Fichero de Comandos Iniciales

El requerimiento principal del sistema diseñado era que tenía que ser capaz de analizar un fichero *log* capturado con la herramienta Snoopy [1].

Por ello, para poder llevarlo a cabo, lo primero que tuvimos que hacer fue ver qué tipo de datos almacenaban dichos registros, viendo que el fichero contenía información sobre qué comando había utilizado, qué usuario, con qué parámetros, en qué momento, bajo qué grupo de usuarios, etc.

Esto, nos llevó a pensar qué datos, de todos los proporcionados, podrían ser relevantes de estudio; o dicho de otra manera, qué información podíamos obtener con ellos; llegando a la conclusión de que, si bien había mucha información contenida en los mismos, el sistema no podría realizar informes bajo todos los aspectos representados, por lo que nos encontramos en la necesidad de centrar la lógica de análisis en algún aspecto concreto.

Todos los datos registrados en el fichero de *logs* que se quería analizar, complementaban un único hecho concreto, y en definitiva, principal: el uso de un comando Linux determinado, por un usuario específico, en un momento concreto, bajo un perfil establecido. Por lo que concluimos que el aspecto principal a analizar del fichero de *logs* proporcionado, sería el propio comando usado y toda la información referente a su utilización.

De ahí la creación del fichero de Comandos Iniciales, un fichero que contendría el diccionario de todos los comandos Linux contemplados y una serie de parámetros de medición de los mismos, de manera que, en conjunto, constituyeran la lógica de análisis del sistema.

Para ello, y teniendo en cuenta que el fichero que queremos analizar recoge únicamente comandos Linux y que hay miles de comandos de este tipo, se creó un programa (*script*) que obtuviese de forma automática una gran colección de comandos existentes directamente del propio “*man*” de Linux, de manera que gracias a la información ahí contenida, fuésemos capaces de sacar información sobre todos y cada uno de los comandos contemplados y pudiésemos generar una lógica de análisis bastante completa para nuestro sistema.

De esta manera, y teniendo en cuenta el programa desarrollado para la obtención de los datos, podemos decir que los pasos seguidos para la constitución de nuestro fichero de Comandos Iniciales ha sido el siguiente:

- 1) Crear un fichero de texto plano con todos los comandos existentes en el sistema.

Para ello, se utilizó la siguiente instrucción directamente desde la línea de comandos de Linux:

```
'compgen -A function -abck > all_commands.txt'
```

De manera que obtuvieses en el fichero de texto *all_commands.txt* todos los comandos disponibles en el sistema Linux.

- 2) Para cada uno de estos comandos, el script desarrollado ejecutaba las siguientes instrucciones con el fin de obtener todos los aspectos del comando contemplado que iban a dotar de lógica a nuestro sistema y que por tanto tenían que estar presentes en nuestro fichero de comandos iniciales:

- Para sacar la descripción y la página del “man” de Linux en la que se encuentra el comando (para nosotros la descripción del comando y su categoría):

`'whatis <<nombreComando>> -l'`

- Para sacar el número de opciones o parámetros que admite cada comando (para nosotros el número de parámetros del comando):

`'man <<nombreComando>> | egrep \"^[:blank:]]{0,8}-\" | wc -l'`

- Para sacar el número de palabras que contiene la página completa del “man” de Linux referente al comando en concreto (para nosotros, la complejidad del comando):

`'man <<nombreComando>> | wc -w'`

- Para sacar la dificultad, se ha utilizado la siguiente fórmula en función del impacto anteriormente descrito (para nosotros, la dificultad del comando):

$$Dificultad = \frac{LOG(complejidad) - MIN(LOG(complejidad))}{MAX(LOG(complejidad)) - MIN(LOG(complejidad))} \times 10$$

Se ha considerado sacar la dificultad a partir de la complejidad, puesto que entendemos que cuantas más palabras se necesiten para completar la explicación del comando, mayor dificultad tendrá el uso del mismo.

- 3) Rellenar el fichero Excel de Comandos Iniciales con los datos obtenidos del programa.

El formato elegido para el fichero de comandos iniciales, ha de respetarse siempre, pues la aplicación está implementada específicamente para tratar los datos de esta forma específica.

Si un usuario “administrador” deseara cambiar el formato de los datos recogidos en este fichero, y/o añadir nuevas columnas, debería hacer los cambios oportunos en el código para que la aplicación sea capaz de leer e interpretar los datos almacenados en el fichero de forma correcta.

✓ **Lectura de Fichero y almacenamiento de los datos**

Un aspecto muy importante de esta aplicación, es que va a tratar con ficheros de *logs* que por lo general tienen un gran tamaño y volumen de datos, por lo que la lectura de los mismos, ha sido un factor muy importante a tener en cuenta.

Para que la aplicación no tardase mucho en leer de fichero (en el caso del fichero de comandos iniciales anteriormente mencionado (un Excel)) se ha desarrollado esta tarea de manera que se minimice al máximo el número de accesos de lectura al mismo.

Para el fichero de *logs*, puesto que su tamaño está alrededor de los 24 MB, hemos decidido hacer la lectura de los datos en varias pasadas, de manera que podamos ir guardando la información contenida en el mismo de forma masiva (con PL's) y así evitarnos realizar accesos constantes a base de datos, y por ende, ralentizar el proceso de lectura del fichero.

En el caso del fichero de comandos iniciales, puesto que el tamaño es sustancialmente menor al del fichero de *logs* que nos han dado para analizar (sólo ocupa 142 KB), nos ha bastado con realizar una única lectura del mismo, almacenando temporalmente todos los datos recogidos en un único objeto compuesto con toda la información contenida en el mismo. De esta forma, hemos conseguido reducir sustancialmente el tiempo de lectura de fichero y agilizar en conjunto toda la funcionalidad.

✓ **PL's para la Inserción Masiva de Registros**

Una vez que hemos leído toda la información contenida en los ficheros, según las especificaciones del punto anterior, hemos utilizado PL's de inserción para la carga masiva de los datos en la base de datos del sistema.

Esto se ha hecho así, para evitar abrir y cerrar conexiones constantemente con la base de datos, y para rentabilizar al máximo las conexiones que se produzcan, reduciendo sustancialmente el tiempo de almacenamiento de los datos.

✓ **Elección del Driver de conexión entre la aplicación y el SGBD**

Un factor que hemos tenido muy en cuenta a la hora de diseñar el sistema, es que el acceso a base de datos debía ser lo más ágil posible, ya que la funcionalidad principal de la aplicación se basa principalmente en el almacenamiento de la información en la base de datos, para su posterior consulta y análisis.

De ahí, que la elección del driver de conexión entre la aplicación y el sistema gestor de la base de datos, haya sido crucial.

Los drivers que tiene el entorno de desarrollo con el que estamos trabajando (.NET) para conectar con Oracle [23], no son tan eficientes como los que existen en otros lenguajes (Java [14], por ejemplo) por lo que para el desarrollo del sistema actual, hemos utilizados dos alternativas distintas, con el fin de sacarle el máximo rendimiento posible:

- Driver ODP (Oracle Data Provider [23]) .NET.

- Permite ejecutar PL's con parámetros.
 - Lo que nos permite poder insertar datos de forma masiva (varios registros en una única llamada).
- Permite capturar los resultados devueltos por un PL que ejecuta una consulta.
 - Esto nos permite poder guardar directamente en Base de datos las consultas complejas con la lógica de análisis de cada tipo de informe que se va a poder realizar con el sistema.
- Driver genérico OleDb.
 - Más rápido para inserciones, borrados, modificaciones y consultas simples.

Viendo las ventajas y los inconvenientes de cada driver, y dada la naturaleza de la aplicación, se ha decidido utilizar ambos drivers de forma conjunta, de forma que aprovechemos las ventajas de cada uno en función del tipo de llamada que se haga en cada caso al SGBD.

Esto nos ha permitido agilizar el tiempo de acceso y consulta a los datos, y sobre todo, el tiempo que tarda la aplicación en procesar el almacenamiento de la información en la base de datos del sistema.

✓ **Uso de una librería para diversas funcionalidades**

Para ciertas funcionalidades del sistema, se ha utilizado una librería propia que aísla a la aplicación del código a bajo nivel encargado de, por ejemplo, grabar los *logs* producidos durante la ejecución del sistema y gestionar la conexión con la base de datos.

En concreto, esta librería proporciona lo siguiente:

- Generación de *logs* del sistema.
- Acceso a base de datos (permitiendo establecer el driver por configuración).
- Envío de emails.

✓ **Implementación de un patrón Observer con la clase BackgroundWorker**

Cuando se envió por primera vez desde la interfaz una solicitud para realizar una tarea que requería cierto tiempo de procesamiento, se vio la necesidad de ejecutarla en un hilo o *thread* diferente al de la propia interfaz, ya que ésta se quedaba 'congelada' durante la ejecución.

Además, puesto que el tiempo de procesamiento del sistema era diferente según el equipo donde se ejecutara la aplicación, y según el tamaño de los ficheros que se le pasasen en la carga, se consideró de vital importancia informar al usuario de la aplicación del progreso de las tareas de misma, sobre todo, de aquellas tareas que pudiesen llegar a tardar un tiempo considerable según las condiciones.

El objeto `BackgroundWorker` de .NET aporta todos estos requerimientos, implementando un patrón `Observer` donde podemos registrar a la propia interfaz para que vaya recibiendo las actualizaciones de progreso y/o estado que la tarea considere oportunas.

✓ **Implementación de un patrón Singleton en la clase `ChargeData`**

Durante el desarrollo de la aplicación y con el fin de conseguir una mayor modularización de las funcionalidades requeridas, nos hemos encontrado con la necesidad de abstraer en una única clase gran parte de los métodos de acceso a la base de datos del sistema, de manera que todos los módulos principales del subsistema de operaciones, tienen que acceder al subsistema datos a través de ella.

Esta clase, es `ChargeData`; y puesto que su funcionalidad es común a todos los módulos que acceden a ella, bastará únicamente con tener una única instancia de la misma por ejecución, de manera que todos la compartan a través del patrón Singleton implementado en la misma.

Este patrón, proporciona la posibilidad de tener una única instancia de la clase `ChargeData` para uso compartido por todos los objetos que la requieran, y más concretamente para proporcionar, en el caso de la inserción masiva de registros, un espacio de almacenamiento en memoria que nos permite ir acumulándolos consecutivamente hasta que llegue el momento de insertarlos todos en base de datos con un único acceso a la misma.

✓ **Inclusión de una librería para internacionalizar los textos de los mensajes de las ventanas de aviso de la interfaz.**

Con el fin de conseguir que la aplicación implementada pueda operar completamente en dos idiomas distintos, inglés y español, cumpliendo así uno de los requisitos marcados por el cliente, se ha incurrido en la necesidad de crear un componente completo para internacionalizar los mensajes generados directamente por la aplicación, de manera que éstos no interfieran con los mensajes proporcionados por los propios formularios de la misma.

Esta internacionalización se refiere principalmente al cambio del idioma en los botones y mensajes de los elementos *Messages Box* propios de .Net, adecuándolos al idioma seleccionado por el usuario para la ejecución de la aplicación.

✓ **Gestión de la navegación entre formularios utilizando el patrón Singleton.**

Para guardar la referencia a cada formulario y no tener que arrastrarla entre ellos cada vez que queremos permitir la navegabilidad, se ha creado un componente con el patrón Singleton que es capaz de conseguir esta funcionalidad.

Este componente (clase *FormReferences*), será por tanto el encargado de crear nuevos formularios en el caso que sea necesario, o de proporcionar la referencia a uno ya existente, evitando que se queden formularios “zombies” sin matar ocultos por la aplicación.

✓ **Utilización de la librería de manejo de ficheros Excel para la generación de los informes.**

Para poder cubrir uno de los requisitos solicitados por el cliente, en concreto, obtener los informes en formato Excel y PDF, hemos utilizado una librería proporcionada por Microsoft que nos permite trabajar con ficheros del paquete office; en nuestro caso *Microsoft.Office.Interop.Excel*, que es la que permite crear, abrir, modificar, cerrar, exportar a PDF, etc. ficheros propios de la herramienta Excel.

✓ **Intento fallido de conexión directa con Google Drive**

En principio se planteó la opción de proporcionar al usuario los informes directamente en el espacio de su Google Drive, pero por cláusulas de seguridad del propio Google y otros factores que condicionaban el correcto funcionamiento del sistema y sobre todo, su simplicidad de uso, se descartó esta opción principalmente por dos motivos:

- Autenticación con Google.
 - Actualmente la autenticación desde código contra Google Drive es bastante compleja, y según hemos podido comprobar, suele ir cambiando con el tiempo en función de las especificaciones que el propio Google considere oportunas, lo que dejaría a la aplicación en *manos* de un agente externo totalmente independiente e incontrolable.
- Configuración de la cuenta de Google Drive.
 - Para poder trabajar desde código con una cuenta Google Drive destino, es necesario configurar ciertos parámetros de la misma de forma totalmente manual, por lo que no podríamos subir ficheros a una cuenta cualquiera especificada por el usuario, sin que ésta hubiese sido anteriormente modificada en base a este requerimiento.

Dados estos problemas se decidió optar por el envío de correos con los informes adjuntos, ya que la conexión con un servidor SMTP es más sencilla y “universal”.

Además, teniendo en cuenta que el requisito inicial del cliente era simplemente poder tener los informes proporcionados por el sistema en el espacio Google Drive de una cuenta determinada, se habló con él directamente y se le especificaron los problemas encontrados para tal fin, dando como solución totalmente factible, cambiar el requisito inicialmente marcado por uno que especificara únicamente la necesidad de dotar a la aplicación de una opción que permitiese el envío de los informes generados por el sistema a una dirección de correo electrónico especificada por el usuario, de manera que si éste proporciona una dirección de Gmail, será el propio Google quien le ofrezca la posibilidad de guardar los adjuntos recibidos en el espacio Google Drive de la cuenta en cuestión, consiguiendo la funcionalidad inicialmente buscada.

7. Implantación

Para conseguir una correcta implantación del sistema desarrollado, se ha elaborado un manual detallado de despliegue (instalación y configuración) del mismo.

Dicho manual, se ha incluido como apéndice al documento actual, por lo que podrá encontrarlo en las páginas finales del presente documento.

8. Evaluación

La evaluación del sistema propuesto se ha llevado a cabo desde dos puntos de vista distintos: desde el punto de vista técnico, y desde el punto de vista de usuario.

Para la evaluarlo desde el punto de vista técnico, se han realizado las pruebas especificadas en el apartado [5.7 Especificación Técnica del Plan de Pruebas](#), que nos permiten comprobar el correcto funcionamiento del sistema en conjunto, la completitud de sus funcionalidades requeridas y el correcto acoplamiento y funcionamiento de cada uno de sus componentes de manera individual, corrigiendo directamente todos los problemas encontrados durante su evaluación.

Para evaluar el sistema propuesto desde el punto de vista del usuario, se ha contado con la colaboración de un par de personas “*testers*” a las que se les ha proporcionado el sistema en cuestión y se les ha explicado, únicamente, la funcionalidad general del mismo.

Con esta herramienta instalada y sabiendo únicamente su funcionalidad principal, se busca que los usuarios nos cuenten su experiencia y satisfacción en el uso de la aplicación, dándonos información de todos aquellos aspectos que considerasen oportunos mejorar.

Fruto de esta experiencia, surgieron diversas mejoras y/o actualizaciones (ya incluidas en la versión final del sistema), pudiendo resumir el conjunto de ellas en los siguientes puntos:

- Adaptación de los informes en base a que tuvieran una lectura más *cómoda* e *intuitiva*.
- Permitir al usuario poder agrupar el listado de informes ofrecidos por la aplicación en torno al aspecto principal de análisis de cada uno de ellos: informes por usuario, informes por comando, informes por categoría, informes generales, o todos.
- Diversas mejoras de aspecto y funcionalidad de la interfaz.
- Adaptar los mensajes de la aplicación a un lenguaje menos técnico, de manera que cualquier persona pueda entenderlos.

9. Planificación y Presupuesto

9.1 Planificación

Con el objetivo de planificar de forma más correcta la duración del proyecto, se ha dividido el mismo en varias fases o etapas, siendo las principales las que se comentan a continuación:

Fase de Análisis

- Estudio del formato de ficheros de Snoopy Logger y qué información podemos sacar de los mismos.
- Creación de casos de uso y diagramas de secuencia.
- Análisis de distintas formas de obtención de informes.
- Definición del modelo de datos.
- Definición de requisitos de usuario y software.

Fase de Diseño

- Diseño del formato admitido para el fichero de comandos, y la información que vamos a almacenar en el mismo, que servirá como base para la lógica de ciertos informes.
- Diseño del formato específico del fichero de *logs*.
- Diseño de la arquitectura de la aplicación (*logs*, datos, envío de emails).
- Diseño de componentes y capas de la aplicación.

Fase de Implementación

- Creación de la base de datos.
- Elección del/os driver/s de conexión con la base de datos.
- Optimización del rendimiento de la carga masiva de *logs* al importar un fichero con un volumen elevado de registros.
- Desarrollo de la arquitectura para el registro de *logs* de la propia aplicación, el acceso a datos y el envío de emails.
- Desarrollo del componente de presentación de la aplicación en .NET (prototipo).

- Desarrollo del componente de datos, con un buffer interno para la carga masiva de *logs*.
- Desarrollo del componente de operaciones, utilizando varios patrones de diseño (Observer, Command, Singleton, etc.) para la ejecución en diferentes hilos de procesamiento de las tareas de carga de comandos y *logs* en la base de datos, de forma que sea posible ir informando del progreso en la interfaz y ésta no quede ‘congelada’.
- Creación de los distintos tipos de informes que existirán en la aplicación:
 - Interfaz común de generación.
 - Filtrado de los mismos mediante usuario, comando y/o categoría.
 - Extracción de toda la configuración de los informes a un modelo de datos y una plantilla Excel.
 - Creación de procedimientos en Oracle [23] para la extracción de los datos de cada informe, con los filtros que sea posible aplicar en cada caso.
- Información general de la BBDD en la pantalla principal de la aplicación: conexión, datos, comparación de comandos vs utilizados.
- Generación y envío de informes por correo electrónico.

Fase de Pruebas

- Pruebas de carga de comandos y *logs*.
- Pruebas de generación y exportación de distintos tipos de informes con diferentes tipos de filtrado.
- Pruebas de borrado de la base de datos.
- Gestión de excepciones en distintos casos extremos (sgbd caído/sin conexión, ficheros de entrada con formato o datos no válidos, etc.)

Fase de Documentación

- Generación de toda la documentación del proyecto de acuerdo a una adaptación de **Métrica v.3 [12]**.

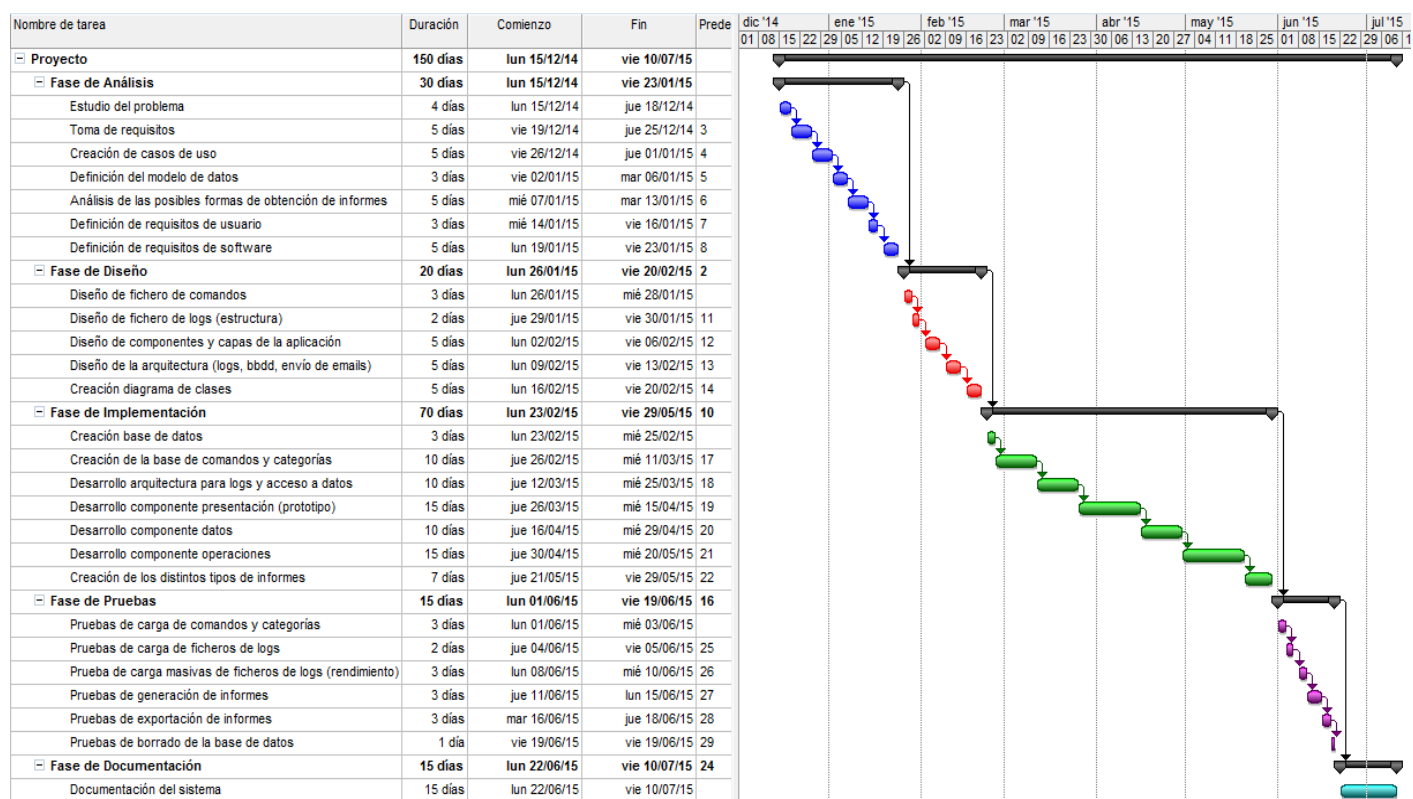


Ilustración 52: Diagrama de Gantt con la Planificación de las Fases del Proyecto

9.2 Presupuesto

En este apartado se describe el presupuesto total utilizado para la realización del presente proyecto, diferenciando los distintos factores que han afectado al mismo.

Para su elaboración, se ha seguido el esquema presentado en la plantilla proporcionada por la universidad para la realización del mismo.

9.2.1 Descripción del Proyecto

Título del Proyecto	Diseño e Implementación de una Herramienta de Análisis de Registros basados en Snoopy para Linux
Duración (Meses)	7 meses
Tasa de Costes Indirectos	20 %

Tabla 162: Presupuesto – Descripción del Proyecto

Los costes indirectos de este proyecto software se refieren principalmente a los costes generales que ha tenido la empresa en cuanto a infraestructura, transporte, gastos de representación, etc. relacionados con el presente desarrollo.

9.2.2 Presupuesto Total del Proyecto

El presupuesto total de este proyecto asciende a la cantidad de TREINTA Y NUEVE MIL TRESCIENTOS CUARENTA EUROS, y tiene una duración de 150 días laborables, 7 meses naturales.

Presupuesto Total del Proyecto (valores en Euros)	39.340 €
--	-----------------

Tabla 163: Presupuesto Total del Proyecto

9.2.3 Desglose Presupuestario (Costes Directos)

Los costes directos de un proyecto software, se refieren principalmente a los costes de personal y materiales (hardware y software) necesarios para la completa consecución del mismo.

9.2.3.1 Costes de Personal

En este apartado, se refleja el coste (en Euros) del personal que constituye el equipo de desarrollo necesario para llevar a cabo el proyecto realizado, distinguiendo cada uno de los perfiles/roles desempeñados durante el mismo.

Únicamente se necesita la colaboración de una persona para cada rol especificado.

Para el cálculo del coste del personal, se han calculado los precios por hora en base a una media realizada entre los salarios ofrecidos en tres empresas de diferentes sectores conocidas por el jefe de proyecto, los publicados en el BOE referentes al XVII Convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos [32] y el coste hombre/mes indicado en la plantilla proporcionada por la universidad para la realización del presente presupuesto.

La jornada laboral para el equipo de desarrollo será considerada de cuatro horas diarias para todos los roles especificados, excepto para el Jefe del proyecto, que si bien estará presente en todas las fases del mismo, su labor principal será la supervisión y apoyo al equipo, por lo que para este perfil, la empresa cliente únicamente computará dos horas de pago de trabajo diarias.

El número de horas de trabajo de cada uno de los roles, vendrá calculado en base al número de horas totales utilizadas en su fase del desarrollo, más su % del tiempo de documentación empleado (siendo este un % proporcional al número de fases utilizadas).

Rol / Categoría	Dedicación (Horas)	Coste (Hombre / Hora)	Coste (Euro)
Analista	132	31	4.092
Diseñador	92	25	2.300
Programador	292	22	6.424
Tester	72	20	1.440
Jefe de Proyecto	306	41	12.546
Total	894		26.802 €

Tabla 164: Costes de Personal del Proyecto

9.2.3.2 Costes de Material

Para la realización del presente proyecto, se ha incurrido en los siguientes gastos de materiales hardware y software:

Descripción	Coste (Euro)	% Uso dedicado	Dedicación (Meses)	Período Depreciación	Coste Imputable ²⁾
Ordenador Sobremesa	1000	100	7	24	291,67
Licencia Software ¹⁾	0	100	7	24	0
Total					291, 67

Tabla 165: Costes de Material del Proyecto

¹⁾ Se ha computado como 0 el valor de las licencias Software utilizadas puesto que éstas corren a cargo del cliente.

²⁾ Fórmula de cálculo de la Amortización: $\frac{A}{B} \times C \times D$

Donde:

- A = nº de meses desde la fecha de facturación en que el equipo es utilizado
- B = periodo de depreciación (meses)
- C = coste del equipo (sin IVA)
- D = % del uso que se dedica al proyecto (habitualmente 100%)

9.2.4 Resumen de Costes

Presupuesto Costes Totales	Presupuesto Costes Totales (Euros)
Personal	26.802
Amortización	291,67
Costes Indirectos	5418,73
Total (sin IVA)	32.512,40
(IVA 21%)	6.827,60
Total	39.340 €

Tabla 166: Resumen de Costes del Proyecto

10. Conclusiones y Trabajos Futuros

En este apartado se presentan las principales conclusiones extraídas de la elaboración del presente trabajo y se plantean posibles vías de desarrollo de la herramienta para dotarla de funcionalidades extra que amplíen su utilidad en el ámbito del análisis de los ficheros log capturados con la herramienta Snoopy [1].

10.1 Conclusiones

Como conclusión principal de este trabajo podemos decir que el desarrollo que se ha llevado a cabo nos ha enseñado cómo el análisis de los ficheros *log* permite a los administradores de sistemas conocer más profundamente el uso que se hace de los equipos informáticos por parte de los usuarios, pudiendo utilizar este conocimiento para potenciar la seguridad en los mismos y adaptar los equipos según quién y cómo se utilicen.

También, hemos podido comprobar cómo el análisis de ficheros *log* es una actividad abierta y en pleno desarrollo, tal y como lo demuestra la multitud de herramientas que se ofrecen actualmente en la red a este efecto. Cada una de las aplicaciones existentes responde a las diferentes necesidades de las personas que la han implementado o de sus clientes, invitando al desarrollo de nuevas aplicaciones con funcionalidades distintas que se adapten a las diversas necesidades que le puedan surgir a cada uno, aportando soluciones parciales e ideas innovadoras que cualquier persona interesada puede aprovechar y potenciar.

En nuestro caso, hemos implementado una herramienta de análisis de logs para un formato de fichero específico capturado con la herramienta Snoopy [1], cumpliendo así con las funcionalidades manifestadas por nuestro cliente y completando el catálogo de las aplicaciones conocidas en este ámbito. Esto, aparte de permitirnos cumplir con los objetivos marcados por el cliente, nos ha dado la posibilidad de intuir en ella un campo de desarrollo promisorio que permitirá al usuario atender cada vez mejor su inquietud de fondo que es: aumentar la seguridad del equipo informático, y a la vez, poder clasificar a los usuarios de los sistemas bajo un perfil tipo marcado en base a los comandos que utilice y al modo en que los use.

10.1.1 Producto

Como conclusión de producto podemos decir que el trabajo realizado ha tenido como fruto el desarrollo (análisis, diseño e implementación) de una aplicación de análisis de *logs* que ayudará al usuario final de la misma a interpretar la información almacenada en los ficheros *log* capturados con la herramienta Snoopy [1], pudiendo aplicar los análisis realizados en pro de una mejora en la seguridad de los equipos y un mejor conocimiento del tipo de usuario que los utiliza.

El producto desarrollado en el presente trabajo es una aplicación que, tal y como se ha descrito en el capítulo del estado del arte de este documento, potencia el conjunto de las herramientas de análisis de *logs* que conocemos actualmente, proporcionando al usuario un sistema con funcionalidades nuevas, capaz de:

1. Analizar un fichero *log* capturado con la herramienta Snoopy [1] y proporcionado por el usuario.

El fichero a analizar siempre ha de cumplir con el formato indicado para esta herramienta.

2. Realizar el análisis de los registros bajo una lógica que podrá ser configurada por el usuario en función de sus propias necesidades.

En este trabajo se ha prestado atención prioritaria a los comandos y a los usuarios registrados, por considerarlos los elementos claves para la necesidad de nuestro cliente; pero esto no evita que la atención pueda extenderse con facilidad a los demás datos registrados en el fichero y almacenados en la base de datos del sistema: fecha de registro, hora de registro, máquina, parámetros utilizados, etc.

3. Permitir al usuario seleccionar el tipo de informe que desea realizar y filtrar la información que quiere obtener.
4. Ofrecer los resultados del análisis realizado en dos formatos distintos: Excel y PDF, dando también la posibilidad de enviar los informes por e-mail a la dirección de correo electrónico que especifique el usuario.

Por lo que podemos concluir que el sistema cumple con todos los objetivos explícitos marcados para la aplicación en la introducción del presente documento y en el estudio de viabilidad del sistema.

Pero, puesto que los registros a analizar constituyen básicamente un listado de todos los comandos utilizados por los usuarios desde la línea de comandos de Linux, y la forma en que éstos han sido utilizados, el análisis de esta información nos ha permitido además poder enriquecer a la aplicación con la funcionalidad de tipificación de usuarios, añadiendo al sistema una base de datos con parámetros relevantes de los posibles comandos a utilizar que permitirá a los administradores del sistema obtener informes en base a los mismos y tipificar a los usuarios en función del manejo que hacen de ellos; de modo que estén en condiciones de prestarles un mejor servicio y de proteger el sistema con mayor efectividad.

En virtud de ello, podemos concluir que la aplicación que presentamos proporciona también a los administradores de sistemas una herramienta capaz de:

5. Ofrecerles la posibilidad de tipificar a los usuarios de los equipos en base a unos criterios de valoración marcados por ellos mismos, de manera que puedan medir la destreza de los usuarios en el manejo de los comandos Linux, dando ellos mismos el peso de cada parámetro de medición del comando en orden a sus necesidades.
6. Configurar nuevos tipos de informes a realizar en función de lo que quieran evaluar en cada momento.

Dicha ampliación de informes se realizará a través de una pequeña actualización en la base de datos, y sin necesidad de tener que tocar el código de la aplicación propiamente dicho.

Estas dos prestaciones, a pesar de estar ya presentes en la actual aplicación, lo están sólo de forma incipiente por lo que nos van a servir de fuente de inspiración para posibles trabajos futuros que nos ayuden a aprovechar mejor su potencialidad en este ámbito.

10.1.2 Proceso

En este apartado, se hace mención de todos aquellos problemas que se nos han dado en la planificación del proyecto, reseñando cómo se han solucionado y qué se ha aprendido de ellos, para concluir después con una serie de conclusiones positivas del proceso del mismo.

Los problemas de planificación que hemos tenido a lo largo del presente desarrollo han sido debidos principalmente a las limitaciones de alcance marcadas al principio del mismo.

Cuando empezamos a desarrollar el proyecto, no conocíamos muy bien el potencial que iba a tener la herramienta, y lo acotamos pensando que su implicación y alcance iban a ser menores de los finalmente conseguidos.

De ahí que según íbamos avanzando en el diseño y análisis de la aplicación, nos íbamos encontrando con nuevas funcionalidades interesantes a cubrir, lo que nos suponía una modificación en el alcance del proyecto y por ende, un nuevo cálculo en la planificación del mismo.

Hasta que no tuvimos bien definido el alcance del proyecto, y no pusimos un límite a todo el potencial que nos íbamos encontrando durante el estudio del mismo, no pudimos dar por cerrada la planificación; lo que nos ha llevado a retrasar la entrega hasta dos meses más de lo pensado inicialmente.

De aquí podemos concluir que en un entorno distinto al académico esto no debería pasar, puesto que los límites del alcance del proyecto deberían estar perfectamente marcados de antemano, y al ser el cliente final quien los marca, han de ser totalmente respetados por el equipo de desarrollo sin dar lugar a nuevas modificaciones y/o ampliaciones que no sean explícitamente solicitadas por el cliente.

Además, durante el proceso de desarrollo del proyecto también nos hemos dado cuenta de cuán fructífero puede ser el trabajo colaborativo entre equipos para el desarrollo de aplicaciones software.

Aunque explícitamente no hemos realizado este trabajo en colaboración con ningún otro equipo desarrollador de herramientas de análisis log, el mero hecho de que ellos tengan colgados en la red sus trabajos nos ha permitido una interacción implícita con ellos, aprovechando las ideas y técnicas presentes en sus trabajos para encontrar una solución más acomodada a las necesidades y especificaciones de nuestro cliente. Evidenciándose, una vez más, la importancia de las comunidades de aprendizaje, de la que la red constituye un ejemplo importante.

10.1.3 Personales

Como conclusión personal, no puedo menos que decir que el trabajo realizado en este proyecto me ha servido principalmente para comprender que todas las asignaturas cursadas en la carrera son importantes.

Para los puntos principales del desarrollo de este proyecto: Estudio de Viabilidad, Análisis, Diseño e Implementación, se ha decidido seguir un estándar estudiado y utilizado en las asignaturas de Ingeniería del Software I, II y III, demostrando así cómo la aplicación de esta metodología española Métrica v.3 [12], especificada principalmente para la construcción de sistemas informáticos, unifica y agiliza el proceso de sistematización de las actividades que dan soporte al ciclo de vida del software y proporciona un marco de gestión para asegurar que los proyectos realizados bajo la misma cumplen sus objetivos en términos de calidad, coste y plazos.

Por otro lado, los conocimientos adquiridos en las asignaturas de Bases de Datos (Fundamentos, Diseño e Implementación, Bases de Datos avanzadas) y Ficheros, han resultado muy útiles para entender el papel que juegan las bases de datos dentro de un Sistema de Información como el nuestro; para realizar los modelos de datos utilizados en el mismo (entidad/relación y relacional), para abstraer y diseñar la aplicación a partir de ellos, y para tener una metodología que nos ayude en el desarrollo, explotación y gestión de los repositorios utilizados para el almacén de datos (*Datawarehouse*).

Gracias a ellas también conocemos las herramientas CASE y el SGBD utilizado en el proyecto, Oracle [23], además del lenguaje SQL utilizado para la realización de las Query's que constituyen los informes y por ende, para el tratamiento y gestión de los datos almacenados y su posterior análisis.

Asignaturas como Estructura de Computadores, Sistemas Operativos y Diseño de Sistemas Operativos, han sido la base principal de este proyecto, ya que gracias a ellas empecé a familiarizarme con el uso de los comandos Linux y su entorno, ayudándome a conocer un sistema operativo hasta entonces totalmente desconocido para mí.

Esta introducción a Linux, también me sirvió de base para poder realizar asignaturas posteriores como Arquitectura de Computadores I y II y Redes I y II, asignaturas que me ayudaron a conocer un mayor número de comandos.

Además, para la implementación de la herramienta propiamente dicha se han puesto en práctica distintos conocimientos adquiridos en diferentes asignaturas de la carrera:

- Análisis y Diseño de Algoritmos
- Programación
- Paradigmas
- Estructura de Datos
- Técnicas de Desarrollo de Programas
- Interfaces de Usuario
- Ingeniería del Conocimiento

Las cinco primeras nos han servido para llevar a cabo la implementación de la aplicación utilizando un lenguaje orientado a objetos como C# [17], en un entorno .Net como Visual Studio [18], teniendo gracias a ellas los conocimientos necesarios para diseñar el problema planteado bajo un paradigma de este tipo de la mejor manera posible, utilizando patrones de diseño como el *Singleton*, el *Observer* o el Modelo-Vista-Controlador, patrón principal utilizado en el diseño de la arquitectura de nuestro sistema.

La asignatura de Interfaces de Usuario nos ha dado las pautas necesarias para diseñar de forma correcta las interfaces de usuario de la propia herramienta; y la asignatura de Ingeniería del Conocimiento nos ha ayudado para la toma, análisis y definición de requisitos en las reuniones con los clientes.

Además, para atender con eficiencia el problema planteado y dar respuesta a lo solicitado por el cliente, no se ha tenido sólo que acudir a los conocimientos adquiridos propiamente del campo de la informática, sino de toda índole, teniendo especial importancia en este sentido aquellas asignaturas que tienen que ver con el análisis lógico, la formulación matemática y la comunicación con las demás personas, en orden a entender bien lo que me estaban pidiendo y expresar lo que se ciertamente se proponía hacer.

Creo que el lenguaje que utilizamos en el campo de la informática, si bien la mayoría de las veces ayuda a comprendernos, otras veces suponen una dificultad que es necesario reconocer para poder afrontar y superar con creatividad y apertura de mente, sobre todo cuando se trata con personas que no están acostumbradas al uso de términos tan técnicos o que no tienen ninguna relación con el campo de estudio en sí mismo.

La realización de este proyecto, a término de conclusión subjetiva y totalmente personal, he de decir que a mí me ha servido mucho para conocer el gran potencial que tiene un sistema operativo, inicialmente tan desconocido para mí, como es Linux; además de hacerme ver la gran importancia y repercusión que puede llegar a tener en nuestra vida el almacenamiento de distintos tipos de registros *logs* y el análisis de los ficheros que los contienen.

10.2 Trabajos futuros

En este apartado se exponen los posibles trabajos futuros que, a mi entender, presenta la herramienta implementada.

Tal y como se ha presentado el proyecto, creo que la aplicación presentada abre camino en torno a tres aspectos distintos, pero a su vez complementarios:

- 1) En torno a la aplicación realizada en sí misma, a fin de perfeccionar lo que se ha hecho y dotar a la aplicación de informes capaces de adecuarse cada vez mejor a sus posibles usuarios.

En este apartado quiero indicar principalmente algunas mejoras que se nos han ido presentando a lo largo del desarrollo del proyecto, pero que por las limitaciones marcadas en el alcance del mismo, no han podido ser tenidas en cuenta para la versión actualmente implementada.

Mejoras como:

- Adaptar a la herramienta para que sea capaz de leer distintos formatos de fichero log

Snoopy es una herramienta totalmente configurable por el usuario, por lo que la aplicación aquí presentada ha sido implementada específicamente para una configuración determinada de la misma, quedando como posible trabajo futuro, la adaptación del sistema a las distintas configuraciones posibles de los ficheros logs capturados, al menos, con esta herramienta.

- Permitir la Carga Acumulativa de Ficheros Log

Actualmente la aplicación únicamente permite cargar un fichero de logs a la vez, por lo que una mejora a proponer sería permitir la carga acumulada de dos o más de ellos, ampliando el campo de análisis de la misma.

- Permitir la Carga Acumulativa de Comandos Iniciales

Actualmente la herramienta sólo es capaz de almacenar un fichero de comandos iniciales a la vez; es decir, que sólo puede interactuar con una lógica de análisis simultáneamente de manera que, cada vez que se carga un nuevo fichero de Comandos Iniciales, se borra toda la información almacenada anteriormente en la base de datos, haciendo que cualquier dato no contemplado en el nuevo fichero y que sí que estuviese en el anterior, se pierda, limitando así el aprendizaje de la herramienta y mermando el potencial de análisis de la misma.

Por ello, una mejora a proponer es permitir almacenar distintas lógicas de análisis a la vez, proporcionando incluso al usuario la opción de elegir cuál de ellas quiere utilizar y dotando así a la herramienta de un mayor potencial de análisis de los logs.

- Permitir la Acumulación de Informes

En esta versión, la herramienta solamente es capaz de proporcionar al usuario el último informe realizado, perdiendo el historial de todos los anteriormente generados, lo que podría mermar en cierta manera el potencial de uso del sistema por parte del administrador, ya que para tener un histórico de los análisis realizados tendría que generarlo él mismo por su cuenta.

- Realizar el Análisis en Tiempo Real

Una ventaja muy a tener en cuenta a la hora de realizar análisis de logs enfocados a la seguridad de sistemas, es el análisis en tiempo real de los logs almacenados, cosa que nuestra herramienta actualmente no es capaz de hacer. Pero, si se le dotara a la misma de este potencial, el ámbito de uso de la misma podría crecer exponencialmente, dando a los administradores de sistemas un arma de mayor envergadura.

- 2) En torno a los temas que la aplicación actual no ha hecho más que insinuar, a fin de extraer de los registros log capturados por la herramienta Snoopy todo lo que de ellos se puede extraer y que pueda ser significativo para el que realiza el análisis.

Con este punto, me refiero principalmente a los distintos ámbitos en los que se podría aplicar el trabajo realizado, ya que en función de lo que se quiere analizar de los registros, podremos aplicar el análisis a estudios y fines diferentes.

En la aplicación implementada, sólo se ha tenido en cuenta un análisis de los ficheros en torno a los comandos utilizados y al uso que se hace de ellos, dejando de lado, por ejemplo, el momento en que los utilizan o desde qué ordenadores acceden.

Si dotásemos a la herramienta de la posibilidad de proporcionar al usuario distintas lógicas de análisis, no sólo la de comandos, le permitiríamos adaptar los análisis realizados con la misma a distintos ámbitos como podría ser la mejora de los equipos en cuanto a prestaciones, y/o la administración de los sistemas en cuanto a las horas que están encendidos innecesariamente, pudiendo utilizar estos espacios, por ejemplo, para realización de *backups* o actualizaciones.

- 3) Entorno a los temas que la aplicación actual permite intuir, y me refiero en concreto, a la tipificación de los usuarios.

Este campo lo considero muy importante en tanto en cuanto creo que dotaría a los profesores y administradores de sistemas armas para mejorar el servicio prestado a los usuarios y para dotar a los sistemas de mayor seguridad. Por ello, su simple introducción en el presente estudio, la considero personalmente como uno de los aportes principales del presente trabajo.

11. Referencias Electrónicas y Bibliográficas

A continuación se enumeran las distintas fuentes de información, definiciones e imágenes que se han consultado y utilizado para la realización del presente documento.

Información General

A continuación se indican las principales fuentes de búsqueda de información que se han utilizado:

- Biblioteca de la Universidad Carlos III de Madrid (UC3M)

<http://www.uc3m.es/ss/Satellite/UC3MInstitucional/es/PortadaMiniSiteC/1371207069408>

- Google

<http://www.google.com>

- Microsoft

<http://www.microsoft.com>

Páginas o Documentos Electrónicos en la Red

En este apartado se indican las principales fuentes electrónicas consultadas para el desarrollo del presente documento (disponibles en Internet):

- [1] Snoopy - Linux Snoopy Logger

<https://github.com/a2o/snoopy>

Fecha de última consulta: [05 / 07 / 2015]

- [2] Lista Top 500

<http://www.top500.org/statistics/list/>

Fecha de última consulta: [05 / 07 / 2015]

- [3] Desktop Operating System Market Share

<http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustommd=0>

Fecha de última consulta: [05 / 07 / 2015]

- [4] Análisis de Ficheros Log
- <http://www.websecurity.es/analisis-los-ficheros-logs>
- Analysis of Execution Log Files
- <http://ieeexplore.ieee.org/strauss.uc3m.es:8080/stamp/stamp.jsp?tp=&arnumber=6062226>
- Analysis of Log Files as a Security Aid
- <http://ieeexplore.ieee.org/strauss.uc3m.es:8080/stamp/stamp.jsp?tp=&arnumber=5974290>
- Graphical Analysis of Computer Log Files
- <http://dl.acm.org/strauss.uc3m.es:8080/citation.cfm?id=198378>
- Human factors in webserver log file analysis: a controlled experiment on investigating malicious activity
- <http://dl.acm.org/strauss.uc3m.es:8080/citation.cfm?id=2600185>
- Fecha de última consulta: [del 1 / 12 / 2014 al 07 / 12 / 2014]
- [5] Event *Log* Explorer
- <http://www.eventlogxp.com/esp/>
- Fecha de última consulta: [08 / 12 / 2014]
- [6] Performance Analysis of *Logs* (PAL) Tool
- <http://pal.codeplex.com/>
- <http://blogs.msdn.com/b/deviations/archive/2008/10/17/pal-performance-analysis-of-logs.aspx>
- <http://blogs.technet.com/b/mikelag/archive/2008/08/20/performance-troubleshooting-using-the-pal-tool.aspx>
- Fecha de última consulta: [09 / 12 / 2014]
- [7] *Log* Parser
- <https://technet.microsoft.com/es-es/scriptcenter/dd919274.aspx>
- <https://upload.wikimedia.org/wikipedia/en/f/fc/Logparser.JPG>
- <http://blogs.msdn.com/b/daniem/archive/2009/04/28/analisis-de-logs-de-iis-utilizando-log-parser.aspx?Redirected=true>

<http://blog.extreme-advice.com/2012/10/30/log-parser-wonderful-yet-under-appreciated-tool-to-read-log-of-windows-iis/>

<https://support.microsoft.com/en-us/kb/910447/es>

<https://technet.microsoft.com/en-us/library/ee692659.aspx>

<http://www.josemanuelmartin.com/2009/03/log-parser-como-explotar-la-informacion-de-tus-logs/>

Fecha de última consulta: [10 / 12 / 2014]

- [8] Log Parser Lizard

http://www.lizard-labs.com/log_parser_lizard.aspx

Fecha de última consulta: [11 / 12 / 2014]

- [9] Visual Log Parser

<https://visuallogparser.codeplex.com/>

Fecha de última consulta: [12 / 12 / 2014]

- [10] Log Parser Studio

<http://blogs.technet.com/b/exchange/archive/2012/03/07/introducing-log-parser-studio.aspx>

Fecha de última consulta: [13 / 12 / 2014]

- [11] Analog

http://fossies.org/windows/www/analog_60w32.zip/

Fecha de última consulta: [14 / 12 / 2014]

- [12] Métrica v.3 - Ministerio de Hacienda y Administraciones Públicas

http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.VSGM0tysVZc

Fecha de última consulta: [23 / 03 / 2015]

- [13] LOPD – Ley de Protección de Datos – Boletín Oficial del Estado

<http://www.boe.es/buscar/act.php?id=BOE-A-1999-23750>

Fecha de última consulta: [23 / 01 / 2015]

- [14] Java – Lenguaje de Programación Orientado a Objetos – Oracle
<https://www.oracle.com/java/index.html>
Fecha de última consulta: [20 / 01 / 2015]
- [15] Eclipse – Entorno de Desarrollo para el lenguaje Java entre otros.
<https://eclipse.org/>
Fecha de última consulta: [20 / 01 / 2015]
- [16] Netbeans – Entorno de Desarrollo principalmente para Java.
<https://netbeans.org/>
Fecha de última consulta: [20 / 01 / 2015]
- [17] C# - Lenguaje de programación orientado a objetos de Microsoft
<https://msdn.microsoft.com/es-es/library/z1zx9t92.aspx>
Fecha de última consulta: [05 / 07 / 2015]
- [18] Visual Studio .Net – Entorno de Desarrollo para lenguajes .Net
<https://msdn.microsoft.com/es-es/library/aa291755%28v=vs.71%29.aspx>
Fecha de última consulta: [05 / 07 / 2015]
- [19] Java EE – Java Enterprise Edition - Oracle
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>
Fecha de última consulta: [05 / 06 / 2015]
- [20] Servidor Apache Tomcat – Contenedor de Servlets para Java Servlet y JSP (Java Server Pages)
<http://tomcat.apache.org/>
Fecha de última consulta: [20 / 01 / 2015]
- [21] PHP – Hypertext Preprocesor – Lenguaje de programación especialmente diseñado para desarrollo web
<http://php.net/>
Fecha de última consulta: [20 / 01 / 2015]

- [22] Microsoft Access –Sistema de gestión de bases de datos de Microsoft.
<https://products.office.com/es-es/access>
Fecha de última consulta: [25 / 01 / 2015]
- [23] Oracle Data Base – Sistema de gestión de bases de datos de Oracle Corporation.
<https://www.oracle.com/database/index.html>
Fecha de última consulta: [25 / 01 / 2015]
ORACLE DATABASE 10G
<http://www.oracle.com/technetwork/es/documentation/317499-esa.pdf>
Fecha de última consulta: [05 / 06 / 2015]
- [24] Precios de las licencias de ORACLE
<http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf>
Fecha de última consulta: [25 / 01 / 2015]
- [25] MySQL – Sistema de gestión de base de datos con licencia dual.
<http://www.mysql.com/>
Fecha de última consulta: [21 / 01 / 2015]
- [26] Precios de las licencias de MySQL
<http://www.oracle.com/us/corporate/pricing/price-lists/mysql-pricelist-183985.pdf>
Fecha de última consulta: [21 / 01 / 2015]
- [27] PostgreSQL– Sistema de gestión de base de datos totalmente gratuito.
<http://www.postgresql.org.es/>
Fecha de última consulta: [22 / 01 / 2015]
- [28] Tipos de requisitos software
<http://www.ie.inf.uc3m.es/grupo/docencia/reglada/psi/unidad6-DOC.pdf>
Fecha de última consulta: [20 / 01 / 2015]

- [29] Definición de Diseño Modular

<http://www.wikiteka.com/apuntes/disenomodular/>

Fecha de última consulta: [15 / 01 / 2015]

- [30] Las 8 reglas de Oro de Shneiderman para el diseño de interfaces de usuario

Título Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva

Autores Ben Shneiderman, Catherine Plaisant

Traducido por Jesús Sánchez Cuadrado, Jesús García Molina

Edición 4ª

Editor Pearson Educación, 2005

ISBN 8420548030, 9788420548036

Nº de páginas 722

- [31] Apuntes para la realización de los Modelos de Datos

<http://ocw.uc3m.es/ingenieria-informatica/disenoyadministraciondebasesdedatos/materialdeclase>

Fecha de última consulta: [13 / 02 / 2015]

- [32] [BOE-A-2013-11199] XVII Convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos.

http://www.boe.es/diario_boe/txt.php?id=BOE-A-2013-11199

Fecha de última consulta: [05 / 07 / 2015]

12. Manual de Implantación del Sistema

Para la correcta implantación del proyecto actual en el entorno donde desee ser utilizado, es necesario seguir los pasos descritos en los siguientes apartados, de forma que al finalizarlos podamos ejecutar y verificar el correcto funcionamiento de la aplicación.

Aunque sería posible la instalación del servidor de BBDD en una máquina distinta a la máquina donde se ejecutará la aplicación principal, en este manual se va a hacer la instalación completa en la misma máquina, si bien se comentará cómo podría configurarse la aplicación para conectarse contra un SGBD instalado en otra máquina distinta.

12.1 Instalación del Sistema Operativo

La versión del sistema operativo que tendremos que tener instalado será como mínimo Windows 7 (cualquier versión de 32/64bits, mientras no aseguremos que el resto de software que instalemos sea para la misma Arquitectura, es decir, tendremos problemas si instalamos Oracle 32 bits y trabajamos con Visual Studio 64bits, por ejemplo).

Para no extendernos en este punto, daremos por supuesto que ya tenemos una máquina lista con el software descrito anteriormente.

12.2 Instalación de Adobe Reader

Podremos descargar de la siguiente dirección una versión gratuita que nos permitirá abrir en modo lectura cualquier documento con la extensión PDF:

<https://acrobat.adobe.com/es/es/products/pdf-reader.html>

No detallamos el proceso de instalación, dada su sencillez.

12.3 Instalación Microsoft Office 2007

Necesitaremos al menos el paquete Excel 2007, ya que sin este Software la aplicación no será capaz de utilizar las librerías de manejo de ficheros Excel.

En este punto tampoco detallamos el proceso de instalación, ya que únicamente necesitaremos tener Excel instalado (da igual los extras que queramos instalar, con el módulo básico nos valdrá).

12.3.1 Parche para permitir exportar archivos de Office como PDF

Además, necesitaremos instalar un complemento para Excel que nos permitirá realizar la exportación de Excel a Pdf en nuestro sistema, y más concretamente desde la aplicación desarrollada.

Este complemento puede encontrarse en la siguiente URL:

<http://www.microsoft.com/es-es/download/confirmation.aspx?id=7>

Instrucciones de instalación:



Gracias por descargar el producto

Complemento de Microsoft Office 2007: Guardar como PDF o XPS de Microsoft

Si la descarga no comienza al cabo de 30 segundos, Haga clic aquí



Instrucciones de instalación

Para instalar esta descarga:

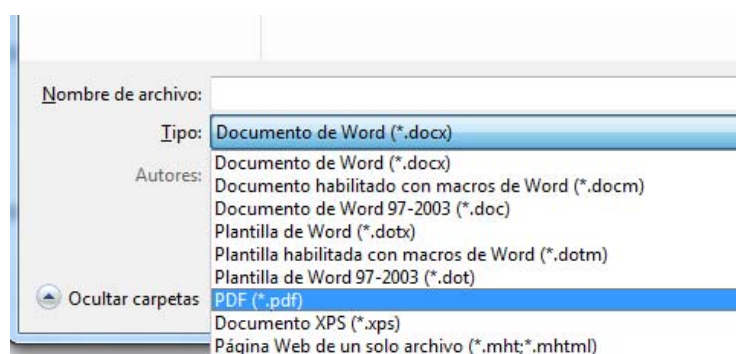
1. Descargue el archivo haciendo clic en el botón **Descargar** (parte superior) y guarde el archivo en el disco duro.
2. Haga doble clic en el archivo de programa **SaveAsPDFandXPS.exe** que se encuentra en el disco duro para iniciar el programa de instalación.
3. Siga las instrucciones que aparecen en pantalla para finalizar la instalación.

Instrucciones de uso:

Una vez que se ha instalado esta descarga, abra el documento que desea publicar y, en función del programa que esté utilizando, seleccione **Guardar o publicar en PDF o XPS** en el menú **Office** o **Archivo**.

Para obtener más información, consulte la Ayuda de Office para saber cómo guardar un archivo en formato PDF o XPS.

Podremos comprobar fácilmente si hemos completado este paso correctamente si al intentar guardar cualquier fichero de Office podemos ver esta opción:



12.4 Instalación de Microsoft Visual Studio .NET 2008

Este paso únicamente será necesario si además de instalar la aplicación también deseamos modificar su código fuente para realizar algún cambio o extensión de la aplicación. Si no es así, podemos saltar directamente al siguiente apartado.

Al abrir el instalador de Visual Studio, se abrirá una ventana con 3 opciones: la primera es para iniciar la instalación de Visual Studio .NET, la segunda es para instalar el MSDN (información, ejemplos de código, ayuda, etc.), y por último el buscador de actualizaciones de servicios de Microsoft. Como sólo necesitamos instalar el programa, hacemos *click* en “Instalar Visual Studio 2008”.



Ilustración 1: Instalación de Microsoft Visual Studio .NET 2008 – Paso I

Siguiendo con la instalación, se abrirá el asistente para la instalación de Microsoft Visual Studio 2008, y la opción la ayudar a mejorar este producto. Hacemos *click* en siguiente para continuar con la instalación.



Ilustración 2: Instalación de Microsoft Visual Studio .NET 2008 – Paso II

En la siguiente ventana tendremos que aceptar los términos de la licencia de Microsoft Visual Studio 2008, además de un formulario para introducir la clave del producto, en caso de que no la coja automáticamente. Aceptamos los términos y pulsamos en Siguiente.

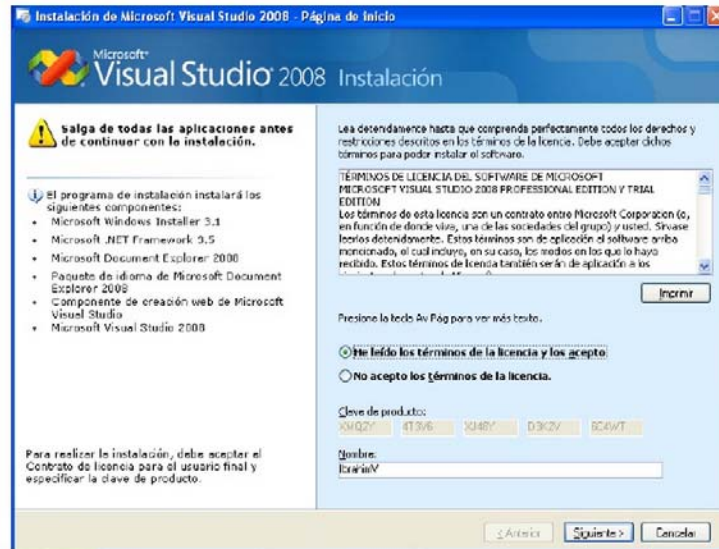


Ilustración 3: Instalación de Microsoft Visual Studio .NET 2008 – Paso III

Seleccionamos la opción de características predeterminadas, la ubicación donde deseemos instalar el programa y hacemos click en “Instalar”.

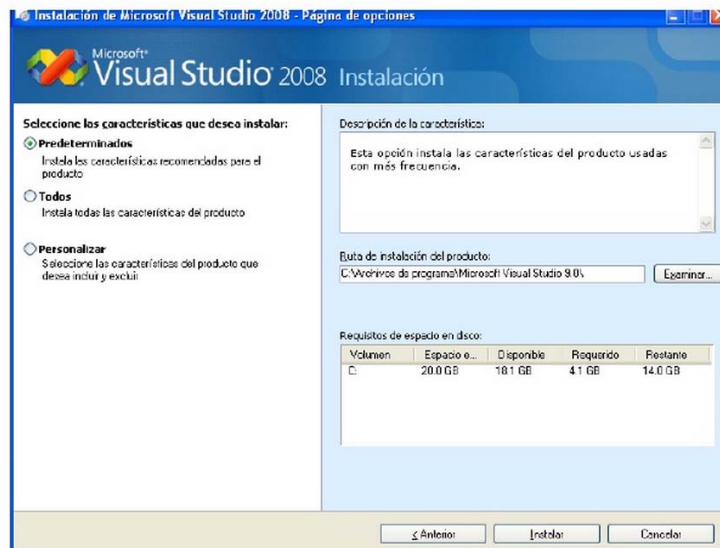


Ilustración 4: Instalación de Microsoft Visual Studio .NET 2008 – Paso IV

Microsoft Visual Studio comenzará a instalarse, así como todos los prerequisites que nuestro sistema necesite para que al finalizar todo funcione correctamente.

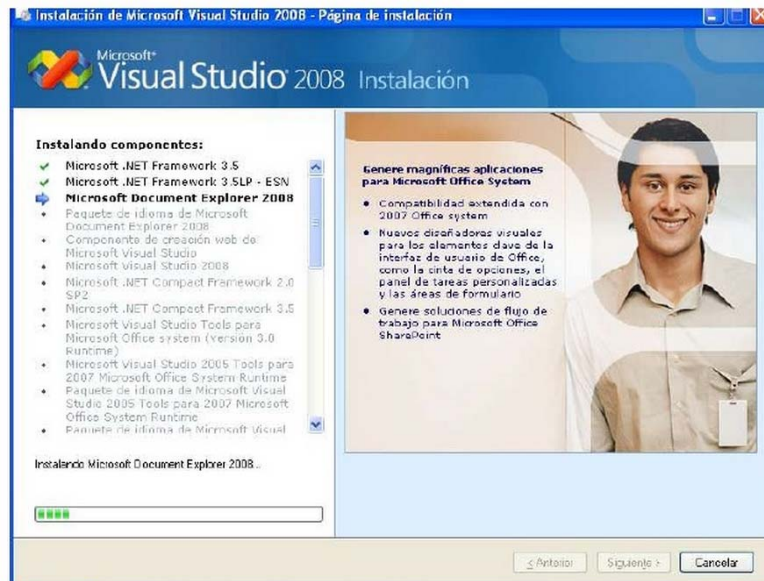


Ilustración 5: Instalación de Microsoft Visual Studio .NET 2008 – Paso V

Una vez completada la instalación, hacemos *click* en “Finalizar”.

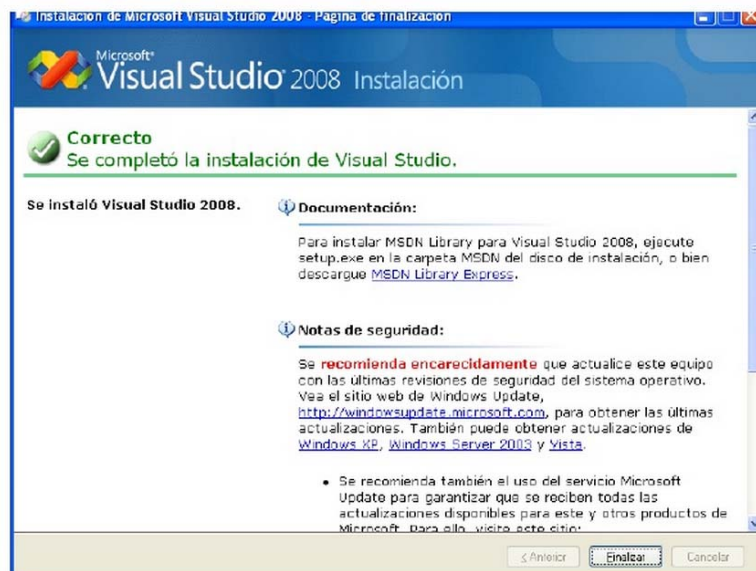


Ilustración 6: Instalación de Microsoft Visual Studio .NET 2008 – Paso VI

Cerramos la ventana que se abrirá a continuación.



Ilustración 7: Instalación de Microsoft Visual Studio .NET 2008 – Paso VII

Y abrimos la aplicación, que nos preguntará por la configuración de entorno predeterminada. Seleccionaremos el entorno Visual C#, ya que este es el lenguaje de programación en el que está desarrollado el proyecto, de entre los existentes dentro del Framework .NET, y pulsamos “Iniciar Visual Studio”.

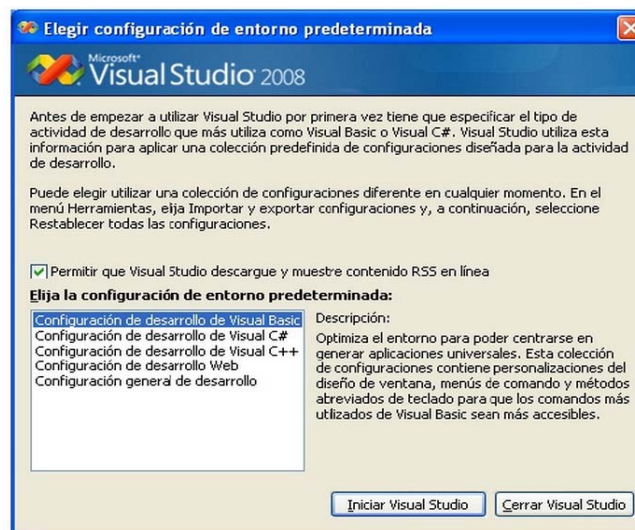


Ilustración 8: Instalación de Microsoft Visual Studio .NET 2008 – Paso VIII

Nos aparecerá una barra de progreso, indicando que está configurando el entorno:

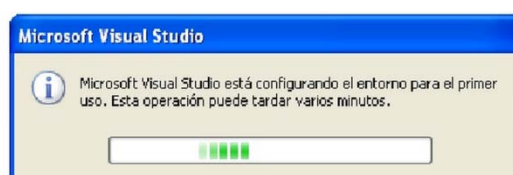


Ilustración 9: Instalación de Microsoft Visual Studio .NET 2008 – Paso IX

Y finalmente se abrirá, mostrando la página de inicio.

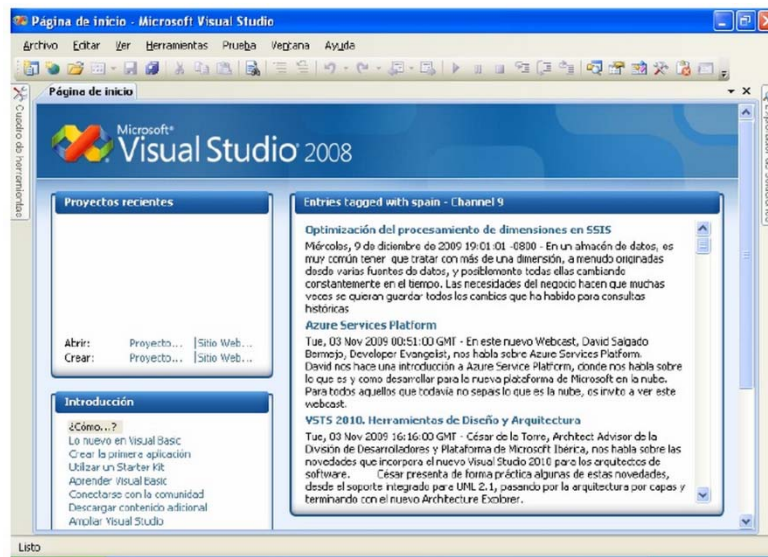


Ilustración 10: Instalación de Microsoft Visual Studio .NET 2008 – Paso X

Ahora ya sólo nos quedaría abrir el proyecto (fichero *.sln), y veríamos las carpetas principales donde está estructurado el proyecto de Visual Studio de este PFC:

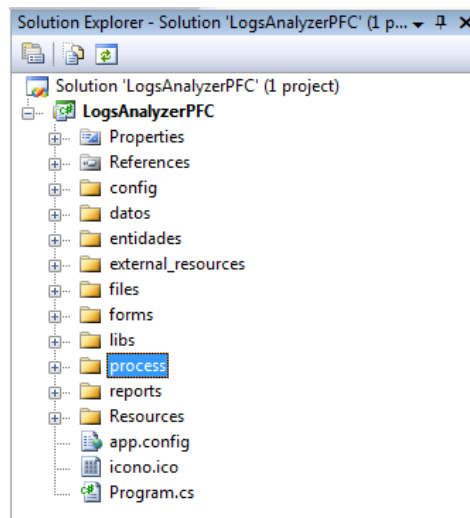


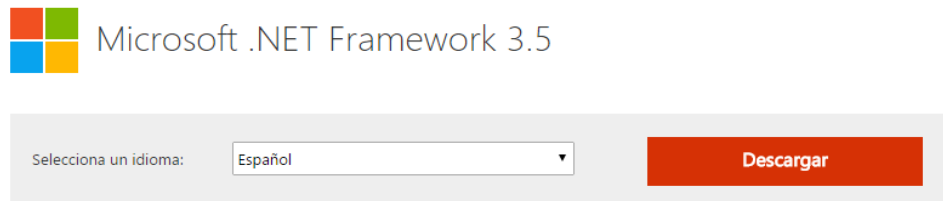
Ilustración 11: Instalación de Microsoft Visual Studio .NET 2008 – Paso XI

12.5 Instalación de Microsoft .NET Framework 3.5

Este paso únicamente es necesario si en el paso anterior decidimos no instalar Visual Studio 2008, ya que si lo tenemos instalado habremos instalado implícitamente también el Framework de .NET.

Aunque suele venir instalado por defecto con Windows 7, para poder arrancar la aplicación necesitaremos tener instalado el Framework de ejecución de aplicaciones de Microsoft, que podremos descargar de la siguiente URL:

<http://download.microsoft.com/download/6/0/f/60fc5854-3cb8-4892-b6db-bd4f42510f28/dotnetfx35.exe>



Microsoft .NET Framework 3.5 contiene muchas y nuevas características que se agregan de forma incremental a .NET Framework 2.0 y 3.0 e incluye .NET Framework 2.0 Service Pack 1 y .NET Framework 3.0 Service Pack 1.

Ilustración 12: Instalación de Microsoft .NET Framework 3.5

Una vez descargado, para instalarlo simplemente abriremos el fichero “dotnetfx35.exe” y seguiremos los pasos de su asistente.

12.6 Instalación de Oracle 10g/11g Express Edition

Podremos descargar esta versión “ligera” de la BBDD de Oracle en la siguiente URL:

<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

Una vez descargada, ejecutaremos el instalador:

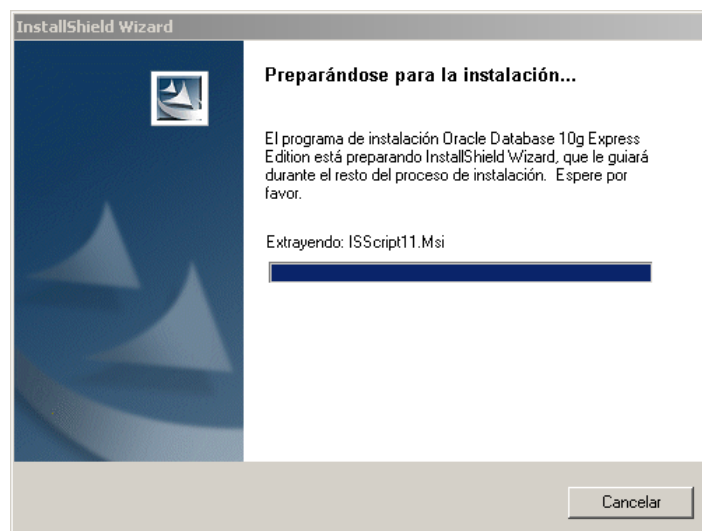


Ilustración 13: Instalación de Oracle 10g/11g Express Edition – Paso I

Pulsamos en siguiente:

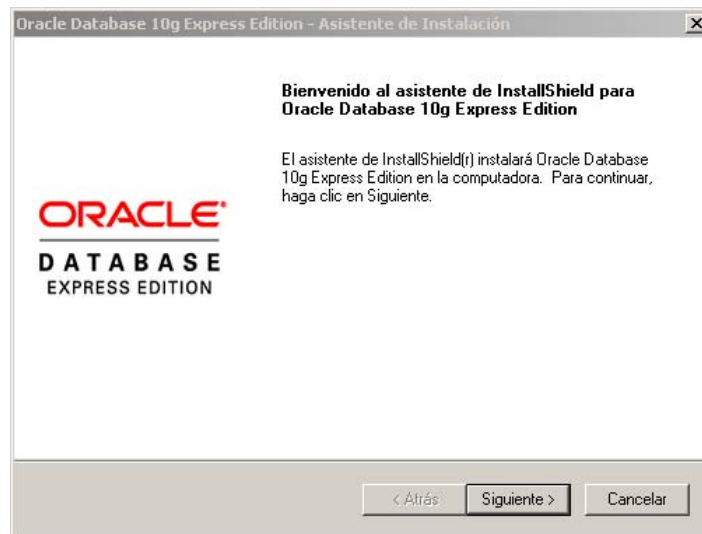


Ilustración 14: Instalación de Oracle 10g/11g Express Edition – Paso II

Aceptamos los términos del acuerdo de licencia:

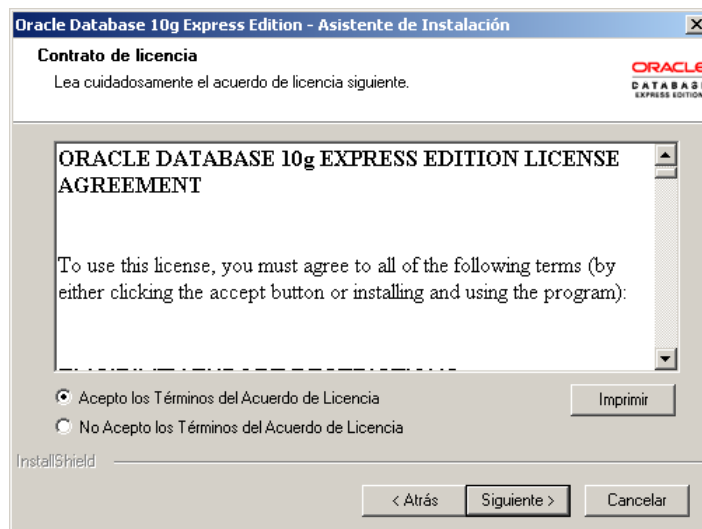


Ilustración 15: Instalación de Oracle 10g/11g Express Edition – Paso III

Posteriormente, se mostrará la carpeta de destino de la instalación, que ubicaremos en la carpeta del servidor donde deseemos que resida la base de datos. Pulsaremos siguiente a continuación:

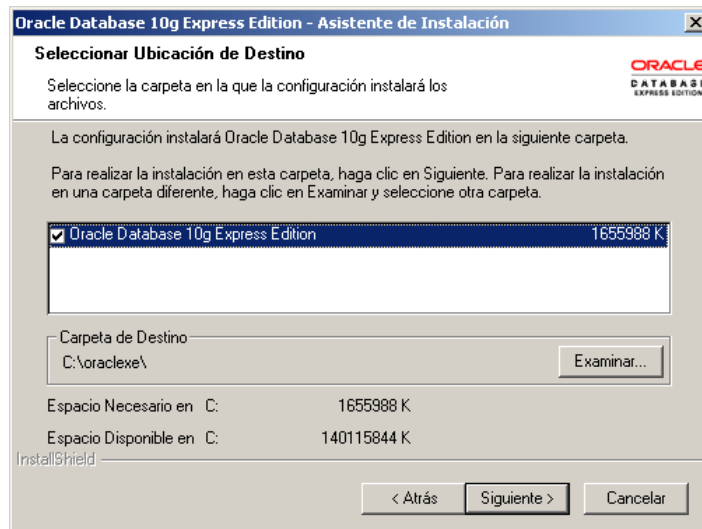


Ilustración 16: Instalación de Oracle 10g/11g Express Edition – Paso IV

Más tarde, se nos solicitará que introduzcamos la contraseña para las cuentas SYS y SYSTEM, que estarán disponibles nada más instalar Oracle. Aquí podemos introducir cualquier contraseña, pero posteriormente nos hará falta para crear una nueva cuenta en la base de datos (para el ejemplo escogimos “pass”, para futuras referencias):

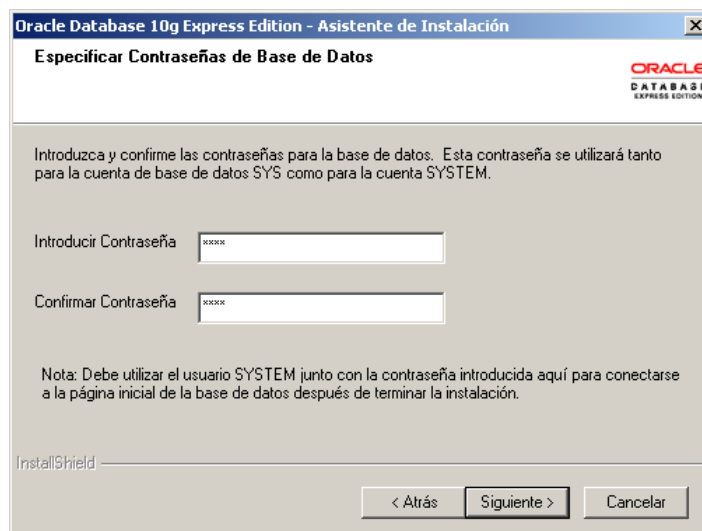


Ilustración 17: Instalación de Oracle 10g/11g Express Edition – Paso V

A continuación, podemos ver los datos de la instalación, justo antes de que comience la misma. Es importante saber que el *Listener* de la Base de Datos de Oracle se arrancará en el puerto 1521, puesto que posteriormente tendremos que configurar nuestra aplicación para que “ataque” a este puerto y de esta forma tengamos conexión con la BBDD.

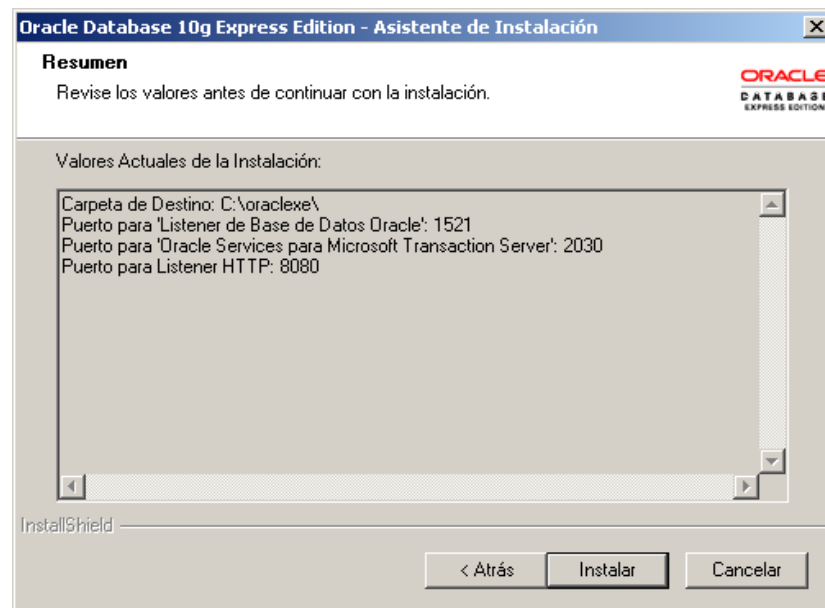


Ilustración 18: Instalación de Oracle 10g/11g Express Edition – Paso VI

Pulsamos instalar, y esperamos a que la instalación termine.

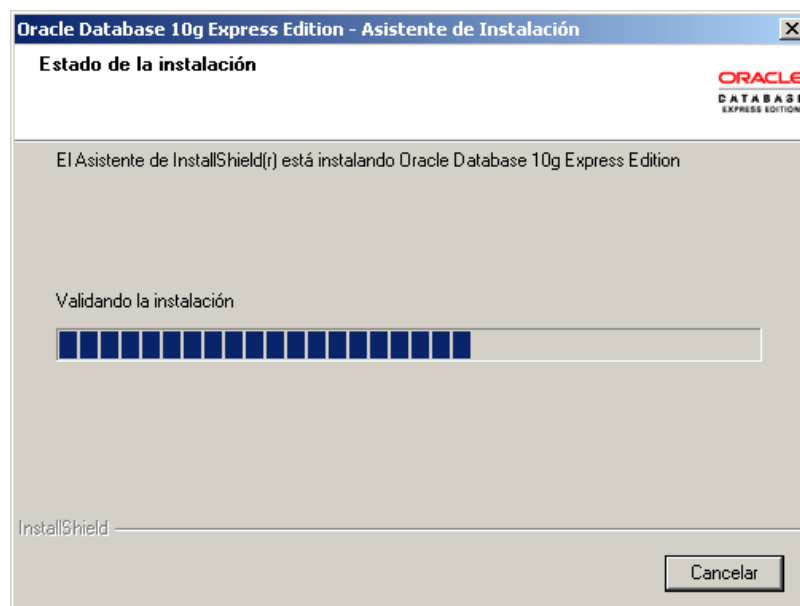


Ilustración 19: Instalación de Oracle 10g/11g Express Edition – Paso VII

Esperamos a que Oracle termine de realizar la configuración inicial...

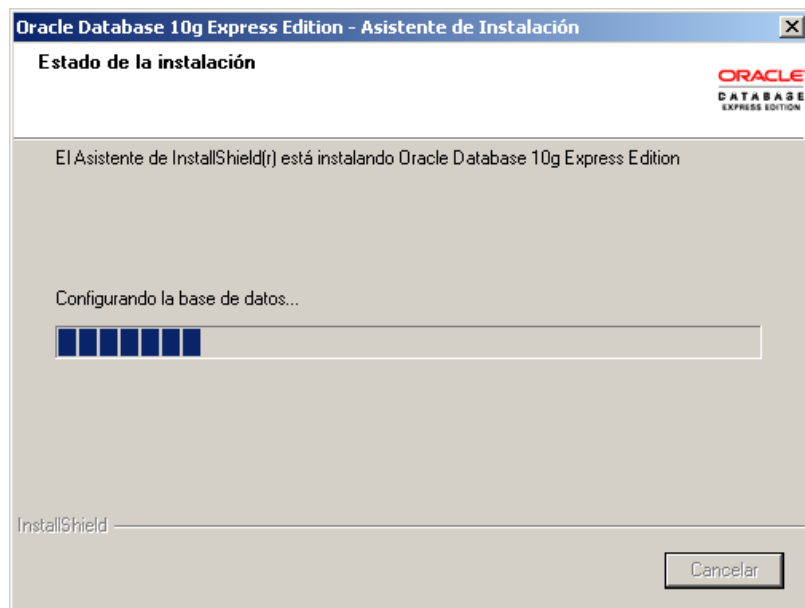


Ilustración 20: Instalación de Oracle 10g/11g Express Edition – Paso VIII

Una vez que el proceso de instalación haya concluido correctamente, se nos preguntará si queremos ir a la página inicial de la base de datos. Marcamos el “check” y pulsamos “Terminar”.



Ilustración 21: Instalación de Oracle 10g/11g Express Edition – Paso IX

Esto nos abrirá la siguiente página:

Ilustración 22: Instalación de Oracle 10g/11g Express Edition – Paso X

Donde podremos logarnos con el usuario SYS/SYSTEM que estará creado con la contraseña que establecimos durante el proceso de instalación.

Una vez nos hayamos logado, tendremos que crear un nuevo usuario en la BBDD, ya que todas las tablas, procedimientos, *triggers*, etc. que crearemos para nuestra aplicación estarán asociados a este usuario en lugar de a uno de los usuarios administradores de Oracle.

Para ello seguiremos los siguientes pasos:

- Acceder a la opción de administración.
- Acceder a la opción de usuarios de base de datos.
- Pulsar el botón de crear.
- Rellenar los datos del nuevo usuario con los siguientes:

Ilustración 23: Instalación de Oracle 10g/11g Express Edition – Paso XI

- Pulsar el botón de crear.

Para verificar si la cuenta fue creada correctamente, podremos cerrar la sesión actual e iniciarla con la nueva cuenta.

12.7 Despliegue, configuración y carga inicial de datos

Además de todos los pasos de instalación de software que se han descrito anteriormente, será necesaria cierta configuración a nivel de la aplicación y de la BBDD, descrita en los siguientes puntos.

12.7.1 Configuración de la cuenta de correo electrónico

Como configuración adicional, en el fichero de configuración de la aplicación (config\pfc.config), podremos modificar los datos de la cuenta de correo electrónico de origen de los correos que la aplicación enviará. Aunque se ha creado una cuenta específica para el proyecto, a continuación se describen algunos pasos especiales que habrá que seguir con una cuenta de correo de Google para que la aplicación pueda conectarse con el servidor SMTP de Google desde código:

```

32: <modulo nombre="email" clase="Arquitectura.Comunicaciones.Email.ModuloEmail" dll="Arquitectura.Comunicaciones.Email.dll">
33:   <parametros-inicio>
34:     <parametro nombre="SmtpServer">smtp.gmail.com</parametro>
35:     <parametro nombre="SmtpPort">587</parametro>
36:     <parametro nombre="Ssl">True</parametro>
37:     <parametro nombre="RequiereCredenciales">True</parametro>
38:     <parametro nombre="FromDefault">aplicacionPFC2015@gmail.com</parametro>
39:     <parametro nombre="FromDefaultPassword">[REDACTED]</parametro>
40:     <parametro nombre="trazas">true</parametro>
41:   </parametros-inicio>
42: </modulo>

```

Ilustración 24: Configuración de la Cuenta de Correo Electrónico – Paso I

Por un lado habrá que activar el “Acceso de aplicaciones menos seguras” en esta URL:

<https://www.google.com/settings/security/lesssecureapps>

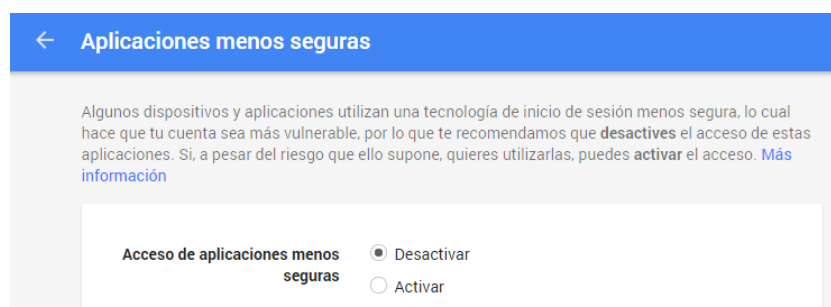


Ilustración 25: Configuración de la Cuenta de Correo Electrónico – Paso II

Y por otro lado permitir el acceso a la cuenta desde nuevos dispositivos:

<https://accounts.google.com/DisplayUnlockCaptcha>

Permitir el acceso a tu cuenta de Google

Como medida de seguridad, es posible que Google te solicite realizar este paso adicional al iniciar sesión en un nuevo dispositivo o aplicación.

Para permitir el acceso, haz clic en el botón Continuar.

Continuar

Ilustración 26: Configuración de la Cuenta de Correo Electrónico – Paso III

12.7.2 Configuración de la aplicación

En el fichero de configuración comentado anteriormente (config\pfc.config) también puede encontrarse información interesante sobre la configuración de la conexión a la BBDD, en concreto los atributos “*ConnectionString*” de estos dos apartados:

```
<modulo nombre="datos" clase="Arquitectura.Datos.ModuloDatos" dll="">
  <parametros-inicio>
    <parametro nombre="SGBD">System.Data.OleDb</parametro>
    <parametro nombre="ConnectionString">Provider=OraOLEDB.Oracle;Data Source=localhost:1521/XE;User Id=root;Password=pass;</parametro>
    <parametro nombre="trazas">true</parametro>
  </parametros-inicio>
</modulo>
<modulo nombre="datosInformes" clase="Arquitectura.Datos.ModuloDatos" dll="">
  <parametros-inicio>
    <parametro nombre="SGBD">Oracle.DataAccess</parametro>
    <parametro nombre="ConnectionString">User Id=root; password=pass;Data Source=localhost:1521/XE; Pooling=false;</parametro>
    <parametro nombre="trazas">true</parametro>
  </parametros-inicio>
</modulo>
```

Ilustración 27: Configuración de la Aplicación – Paso I

Como se puede ver ya están configurados para apuntar a “localhost:1521/XE”, que es donde debería estar arrancado el *listener* de Oracle que se instala por defecto.

Si quisiéramos configurar una base de datos en otra máquina distinta bastaría con poner algo como esto:

“Data Source=//<IP>:1521/XE”, donde <IP> sería la dirección de la máquina donde estuviera instalado Oracle.

12.7.3 Carga de datos

Antes de poder utilizar la aplicación, tendremos que ejecutar los Scripts de base de datos con los que crearemos todas las tablas, procedimientos, *triggers*, e insertaremos los datos necesarios para que la aplicación funcione desde una instalación nueva, como es el caso.

Para este punto bastará con ejecutar los Scripts ubicados en la siguiente carpeta de la aplicación:

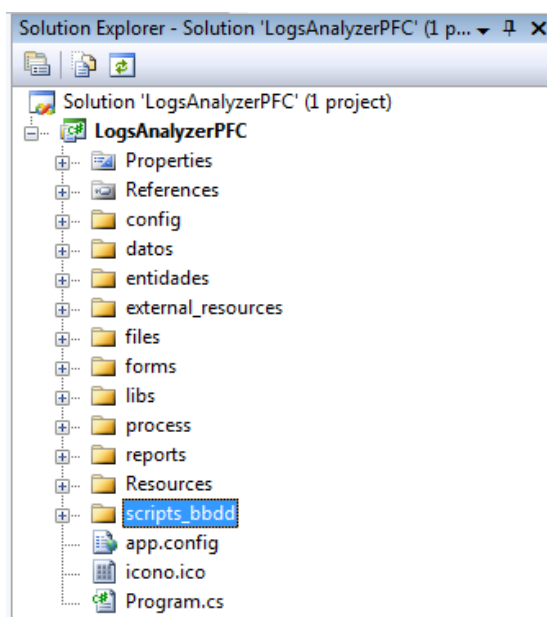


Ilustración 28: Carga de Datos – Paso I

Y para ejecutar dichos Scripts podremos hacerlo a través de la siguiente herramienta, que se habrá instalado dentro de la carpeta de instalación de Oracle:

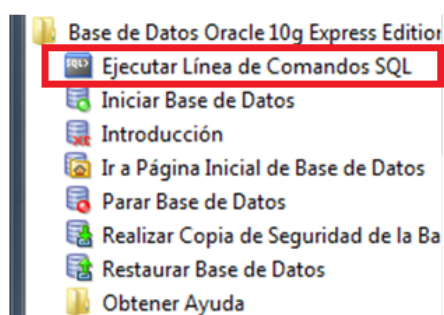


Ilustración 29: Carga de Datos – Paso II

Simplemente nos conectamos con el usuario “root”, y a continuación podemos lanzar un Script SQL a través del “loader” de Oracle, simplemente precediendo a la ruta del SQL del carácter “@”.

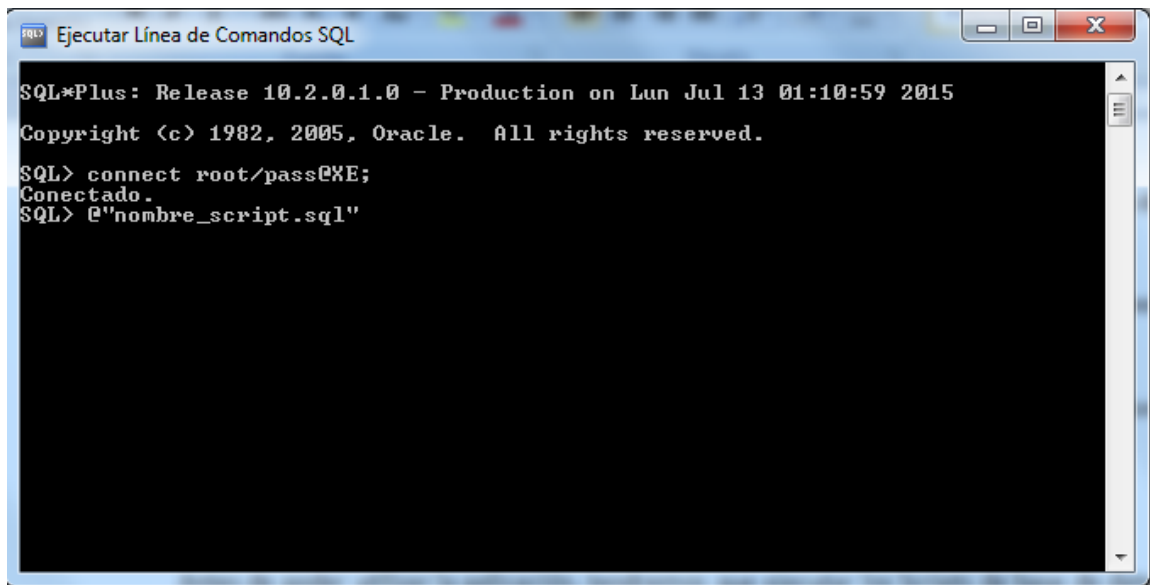


Ilustración 30: Carga de Datos – Paso III

12.8 Procedimientos de verificación

Para verificar que hemos realizado correctamente todos los pasos anteriores, podemos hacer una prueba sencilla con la aplicación, que nos ayudará a detectar si hay algún problema.

En primer lugar abriremos la aplicación a través de su ejecutable:

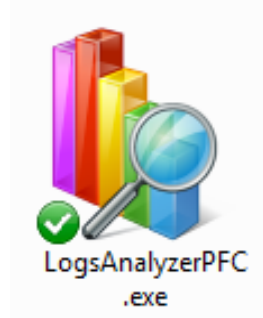


Ilustración 31: Procedimientos de Verificación – Paso I

Podemos encontrar este fichero en la carpeta “\LogsAnalyzerPFC\bin\Debug”.

A continuación deberíamos ver la pantalla principal de la aplicación:



Ilustración 32: Procedimientos de Verificación – Paso II

Lo importante aquí es que el icono de conexión con la BBDD aparezca de color verde, ya que con esto podemos estar seguros de que la aplicación es capaz de conectarse con Oracle.

Una vez comprobado esto, podemos hacer una prueba rápida de importar el fichero de comandos y un fichero de logs (ver manual de usuario en la propia aplicación) para poder comenzar un análisis de datos.

Por último, al abrir la pantalla de generación de informes y seleccionar un tipo de informes a mostrar, deberíamos ver los informes en la parte resaltada:



Ilustración 33: Procedimientos de Verificación – Paso III

Esto indica que la carga inicial de BBDD se ha realizado correctamente, y que la aplicación debería funcionar con normalidad sin necesidad de más configuraciones.

13. Manual de Usuario

El manual de usuario de la aplicación desarrollada, puede encontrarse directamente en la pantalla principal de la misma, en el apartado Información/Guía de Uso.



Ilustración 1: Obtención del Manual de Usuario